DOS EXTRA Nr. 10 – 1990

DOS EXTRA

18,– DM 145 öS 18,– sfr 26,50 Fm

Sonderausgabe DOS EXTRA

Nr. 10 1990

ISSN 0934–2842

DMV-Verlag

Eine Publikation der DOS INTERNATIONAL

Hardware-Praxis

- Festplatteneinbau
- Die Alternative: 80386SX
- Serielle Schnittstelle einfach programmiert

MS-DOS 4.01

- Alle neuen Befehle
- 4.01-Interrupts
- DOS-Shell-Werkstatt

Turbo Pascal

- Wie funktionieren BGI-Treiber?
- Unit-Sammlung zum Abtippen

Zum Abtippen

- Hercules-Grafik für SBasic
- Snapshot 2.7
- TSR-Datei-Manager

Über 10 Seiten Software-Hotline

CPS Computertechnik GmbH
Hamburger Str. 283 · 3300 Braunschweig
Telefax (05 31) 33 06 61

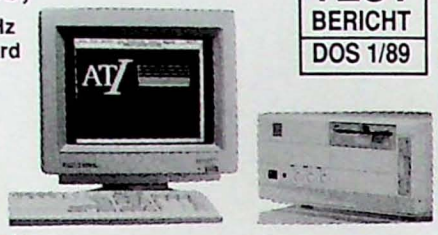
Ladenöffnungszeiten: Mo-Fr 9-13 + 15-18 Uhr
Sa 9-13 Uhr

☎ (05 31) 33 06 63

- 24 Monate Garantie auf CPS-Rechner
- 10 Tage Rückgaberecht auf Hardware
- Eigener Reparatur-Service

AT I — DM 2998,-

siehe TEST BERICHT DOS 1/89



- CPU 80286-12, 0-Wait-State, 12 MHz
- 640 KB RAM erw. bis 4 MB on board
- 1 Laufwerk 5,25", 1,2 MB TEAC
- HDD / FDD Contr. Interleave 1 : 1
- 20 MB Festplatte 38 ms
- 1 ser./1 par./Game-Port
- Herc. komp. Grafikkarte
- Cherry Tastatur MF-II, 102 Tasten
- 14" Monitor s/w
- MS-DOS 4.01 deutsch und GW Basic

386 SX — DM 4175,-

siehe TEST BERICHT DOS-TEST 1/90

- 80386 SX CPU - 16 MHz, 0-Wait-State
- 2 MB RAM, bis 4 MB on board
- Sockel für 80387 Co-Prozessor
- Herc. komp. Grafikkarte mit Printerport
- 1 ser./1 par./Game-Port
- 5,25" Laufwerk 1,2 MB TEAC
- 3,5" Laufwerk 720 KB / 1,44 MB TEAC
- HDD/FDD Contr. Interleave 1 : 1
- 40 MB Festplatte, 28 ms Zugriffszeit
- Cherry Tastatur mit sep. Cursorblock
- 14" Flat Screen Monitor s/w oder amber
- MS-DOS 4.01 deutsch und GW Basic

mit VGA-Grafikkarte und VGA Color-Monitor 14" DM 4955,-

Überzeugende Technik von CPS

AT II

- CPU 80286-16, 0-Wait-State, 12/16 MHz
- 1 MB RAM on board, bis 4 MB aufrüstbar
- 5,25" Laufwerk 1,2 MB
- 3,5" Laufwerk 720 KB / 1,44 MB
- HDD / FDD Contr. Interleave 1:1
- 20 MB Festplatte 38 ms
- ser./par./Game-Port
- VGA-Grafikkarte
- Cherry Tastatur MF-II, 102 Tasten
- 14" VGA-Monitor color
- MS-DOS 4.01 deutsch und GW Basic
- Textomat Plus - Textverarbeitung

Mini-Tower wie Abbildung DM 4560,-
Tisch-AT DM 4360,-

Sämtliche Angebote freibleibend, Zwischenverkauf vorbehalten. Wir liefern an Nicht-Kaufleute nur per UPS-Nachnahme. Ins Ausland nur per Vorkasse. Fordern Sie unseren Gesamtkatalog gegen 3,- DM in Briefmarken. Es gelten unsere allgemeinen Geschäftsbedingungen.

Versand nur über Braunschweig!

DRUCKER

OKI Microline 390 Centr.	Auf Anfr.
OKI Microline 380	1138,-
OKI OL 400 + 800 Laserline	Sonderpreis auf Anfr.
NEC P 6 Plus Centr.	1499,-
NEC Silentwriter LC 890, Postscript, 3 MB RAM	6995,-
Star LC 10	448,-
Star LC 10 II	528,50
Star LC 24-10	680,-
Star XB 24-10	1437,-
EPSON LX 400	448,-
EPSON LQ 400	719,-
EPSON LQ 550	982,-
Centronics Druckerkabel	18,80

Weitere Drucker auf Anfrage

Wir liefern nur mit dt. Handbuch, Seriennummer und Herstellergarantie!!! Drucker-Grauimporte mit engl. Handbuch, ohne Seriennummer, ohne Herstellergarantie sind bei uns ausgeschlossen.

star LC 10

- 9 Nadeln
- 144 Zeichen pro Sekunde
- Papierparkfunktion
- NLQ / LQ

DM 448,-

OKI ML 390

- 24 Nadeldrucker
- 270 Zeichen pro Sekunde
- 90 Zeichen Briefqualität
- Parkfunktion

DM 1383,-

ohne Abbildung

EPSON LX 400	DM 448,-
EPSON LQ 400	DM 719,-
OKI ML 393	DM 2825,-
OKI ML 393 Color	DM 3038,-

COMMODORE AMIGA 2000

DIE PC-GENERATION DER ZUKUNFT

XT-, AT-, Unix-fähig. Verbindung der Prozessorwelten Intel und Motorola. Echtzeit-Multitasking sowie Animation. Unschlagbar in der Kombination Text und Grafik/Farbe. Modular nach individuellen Bedürfnissen erweiterbar. Von Commodore - Marktführer bei Mikrocomputern. Wir beraten Sie gerne ausführlich.

DM 1985,-
ohne Monitor

Alle reden von High Tech - wir produzieren High Tech.

CPS AT III / Cache

- DOS - OS/2 - UNIX - XENIX - fähig
- 80386-25 CPU, 20/25 MHz, 32 KB Cache
- Memory Expansion Card. EMS Funktion
- 0-Wait-State, Sockel für 80387
- 2 MB RAM, erweiterbar bis 16 MB on board
- 2 ser./2 par./Game-Port
- 1 LW 1,2 MB 5,25" TEAC
- 1 LW 1,44 MB 3,5" TEAC
- VGA Grafikkarte EIZO MD-B 10
  1024 x 768 Punkte bei 16 Farben
  800 x 600 Punkte bei 256 Farben
  512 KB Videospeicher, 16 BIT Bus
- SCSI Combi Contr. Datentransfer 10 MByte/sec. Multiuser, Multitasking
  Software Support: DOS 3.x, OS/2, NOVEL/UNIX/XENIX
- 160 MB SCSI Festplatte 17 ms
- 14" Multiscan Monitor EIZO 9060s
- MS-DOS 4.01 dt. und GW-Basic
- UNIX V/386 OS

DM 16.500,-

Bitte schicken Sie mir Prospektmaterial:

NAME

STRASSE

PLZ/ORT

Ausschneiden und einsenden an:
CPS Computertechnik GmbH
Hamburger Straße 283
3300 Braunschweig

DISKETTEN

NN 2DD 3,5" 10 Stück	15,00
NN 2DD 5,25" 10 Stück	6,90
Magix MF 2DD 3,5"	24,50
Select MF 2DD 3,5"	33,50
Select MD 2DD 5,25"	19,00
Fuji MD 2DD 5,25"	27,00

Nutzen Sie unseren Bequem-Kauf-Kredit!

SOFTWARE

ALDUS Pagemaker 3.0	2150,-
GEM Artline 1.0	1390,-
GEM Desktop 1.1	1200,-
MS-DOS Lernprogramm 2.0	168,-
MS-Windows 286 2.1	480,-
MS-Windows 386 2.1	750,-

weitere Software auf Anfrage

FARBBÄNDER

STAR NL/NG/ND/NR-10, Stück	8,60
EPSON LX-800/LQ-500, Stück	7,50
PANASONIC KX-P, Stück	9,25
OKI ML 320	7,20
OKI ML 390	10,40
NEC 2200	10,25
NEC P6+/P7+	10,90
Star LC 10	7,00

Die richtige Mischung macht's

Wieder einmal ist eine DOS Extra fällig, und wieder haben wir uns Gedanken darüber gemacht, was für unsere Leser von besonderem Interesse ist. Das Resultat dieser Überlegungen sind die Themen MS-DOS 4.01, das ja für Gesprächsthema gesorgt hat, die Programmiersprache Turbo Pascal, das mit Abstand liebste Kind der PC-Programmierer, und, einmal von anderer Seite her beleuchtet, die Hardware des PC.

Die Version 4.01 des Betriebssystems MS-DOS hat bei ihrem Erscheinen für Furore gesorgt: Es vollzog geschickt die Verwandlung von der häßlichen Raupe zum schönen Schmetterling. Verzeihen Sie den etwas poetischen Vergleich, aber eben dies war die Intention der Microsoft-Entwickler, die Attraktivität der unscheinbaren Benutzerebene des »alten« MS-DOS durch eine optisch ansprechende und zugleich komfortable Benutzeroberfläche zu erhöhen.

Wie sieht es nun aber in der Praxis mit der berühmten DOS-Shell aus? Dieser Frage haben wir einen ausführlichen Werkstatt-Bericht gewidmet. Auch wenn die DOS-Shell für Sie bisher kein ernstzunehmendes Arbeitswerkzeug war, so birgt sie dennoch viele sinnvolle Funktionen in sich. Des weiteren erfahren Sie, welche neuen Befehle in der DOS-Version 4.01 hinzugekommen sind und welche Einträge sich bei dieser neuen Version für die Config-Datei empfehlen. Daß sich aus dem neuen Befehlsumfang auch neue Möglichkeiten für Freunde von Stapeldateien ergeben, ist eine logische Konsequenz: Welche das im einzelnen sind, verrät Ihnen ein eigener Beitrag. Abschließend haben wir Meister Microsoft höchstpersönlich um eine Stellungnahme gebeten...

Wie ich zu Beginn bereits bemerkte, kennt die Beliebtheit von Turbo Pascal keine Grenzen. Wir haben uns eine Spezialität des Borland-Produkts herausgepickt, die BGI-Treiber. Ein ausführlicher Beitrag behandelt die internen Spezialitäten dieser Grafiktreiber. Was sind BGI-Treiber, wozu dienen sie, wo werden sie eingesetzt? Ein Patch, der einen »firmeneigenen« Fehler dieser Treiber ausmerzt, rundet das Thema ab.

Daß es eine Menge Utilities zu Turbo Pascal gibt, ist bekannt, weniger hingegen, daß es auch in der Shareware-Ecke einiges zu entdecken gibt. Einige besonders interessante Programme stellen wir Ihnen vor. Als Krönung des Pascal-Themas gibt es für die DOS-Extra-Leser drei Units, die sich der Musikprogrammierung, einer verblüffend schnellen Bildschirmausgabe für den Textmodus und der Datumsberechnung widmen.

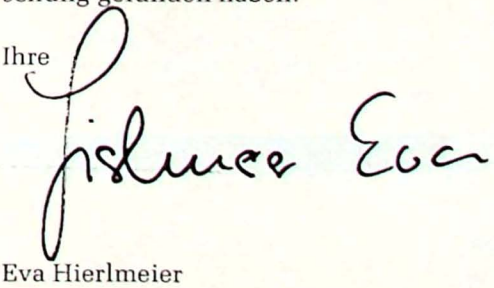
Die Hardware ihres PCs macht vielen Anwendern zu schaffen. Sowohl zu den Schwierigkeiten bei der Festplatteninstallation oder der Konfigurierung der seriellen Schnittstelle oder den Unklarheiten, die über die wirklichen Leistungsvorteile eines 386SX herrschen, geben wir Anleitung und Informationen. Und was das Rätsel um die eigene Hardware-Ausstattung anbelangt: Die klärt ein Listing, das über die Konfiguration und die Leistungsdaten Ihres PC Auskunft gibt.

Inzwischen ist die DOS Extra dafür bekannt, daß sie erstklassige Listings enthält. Natürlich haben wir uns auch dieses Mal bemüht, diesem Ruf gerecht zu werden. Sie begegnen bereits bekannten Namen wie SAM III, der nun in einer weiter verbesserten, fast perfekten Form bei der Version 1.1 angelangt ist. Auch Snapshot ist eine Versionsnummer weiter; 2.7 heißt sie und ist nun endlich nicht mehr an die 64-KByte-Grenze der Bilddateien gebunden. Auch ein neues Bildformat mit 16 Farben in CGA-Auflösung bei 160x100 Punkten gehört ab sofort zum Funktionsumfang. SBasic-Programmierer kommen ebenfalls auf ihre Kosten. Sie können dank neuer Funktionen auch Hercules-Grafiken realisieren, und ein Patch bringt weitere Verbesserungen.

Ein pfiffiges Drucker-Utility kann wohl jeder gebrauchen. »prdat.asm« leitet trickreich Daten in Dateien um. Formatierte Textdateien werden später per MS-DOS-Copy-Befehl bequem an den Drucker weitergeleitet. Ferner gibt es noch ein DOS-Utility, mit dem Sie aus jeder beliebigen Anwendung heraus problemlos alle Dateioperationen ausführen können.

Wir hoffen, daß wir für Sie auch diesmal die richtige Mischung gefunden haben.

Ihre



Eva Hierlmeier

Turbo Pascal

Der Weg zur Unabhängigkeit 6
– *BGI-Treiber*

Shareware-Tools für Turbo Pascal 10

Units-Bibliotheken à la Turbo Pascal 14
– *Sound, Bildschirmausgabe, Datum*

MS-DOS 4.01

MS-DOS 4.0 24
– *Vom häßlichen Entlein zum stolzen Schwan*

Die Datei »autoexec.bat« glasklar 26

Was ist neu in »config.sys« 28

Die neuen Befehle von MS-DOS 4.01 30

Komfort ab Werk 33
– *Die Shell-Werkstatt*

MS-DOS 4.01 – neue Betriebssystemfunktionen 38

PC-Hardware

80386SX – der 386-PC für jedermann 43

Der Draht zur Welt 46
– *Konfiguration der seriellen Schnittstelle*

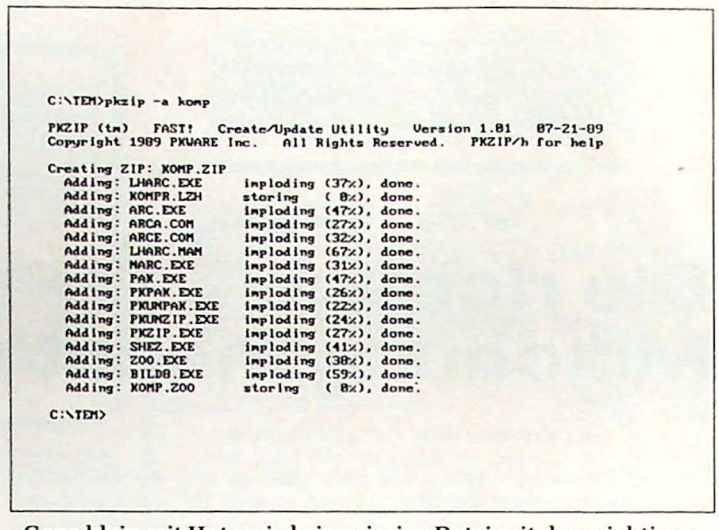
Festplatteneinbau von A bis Z 48

Dem PC auf den Zahn gefühlt 52
– *Prüflisting für Leistungsdaten der Hardware*

Drucker-Lexikon 58

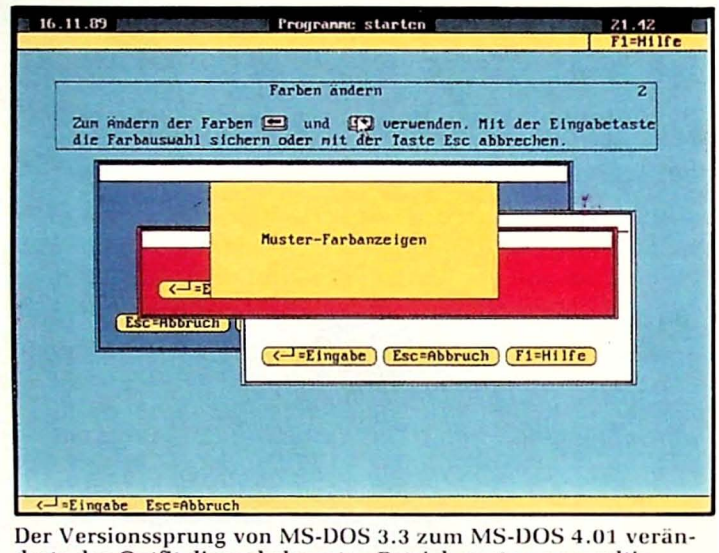
Grundlagen

Datenkompression 60
– *So klein mit Hut*

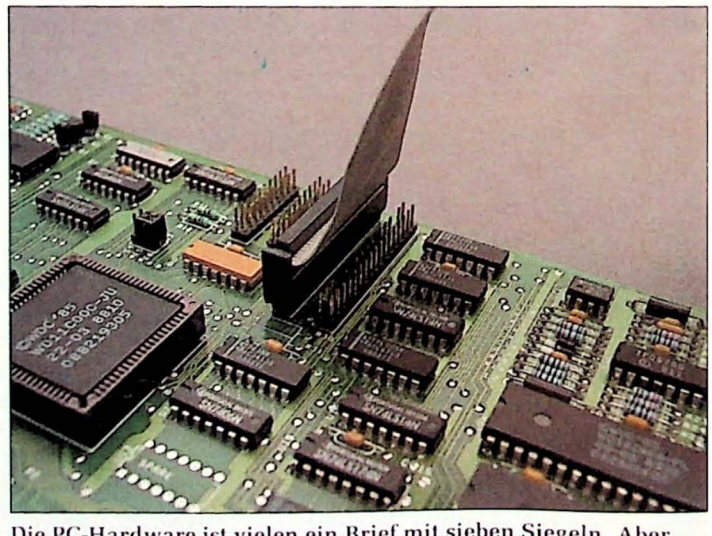


```
C:\TEM>pkzip -a komp

PKZIP (tm)   FAST!   Create/Update Utility   Version 1.01   07-21-89
Copyright 1989 PKWARE Inc.   All Rights Reserved.   PKZIP/h for help

Creating ZIP: KOMP.ZIP
  Adding: LHARC.EXE      imploding (37%), done.
  Adding: KOMPR.LZH      storing   ( 0%), done.
  Adding: ARC.EXE        imploding (47%), done.
  Adding: ARCA.COM       imploding (27%), done.
  Adding: ARCE.COM       imploding (32%), done.
  Adding: LHARC.MAN      imploding (67%), done.
  Adding: MARC.EXE       imploding (31%), done.
  Adding: PAK.EXE        imploding (47%), done.
  Adding: PKPAK.EXE      imploding (26%), done.
  Adding: PKUNPAK.EXE    imploding (22%), done.
  Adding: PKUNZIP.EXE    imploding (24%), done.
  Adding: PKZIP.EXE      imploding (27%), done.
  Adding: SHEZ.EXE       imploding (41%), done.
  Adding: ZOO.EXE        imploding (38%), done.
  Adding: BILD8.EXE      imploding (59%), done.
  Adding: KOMP.ZOO       storing   ( 0%), done.

C:\TEM>
```

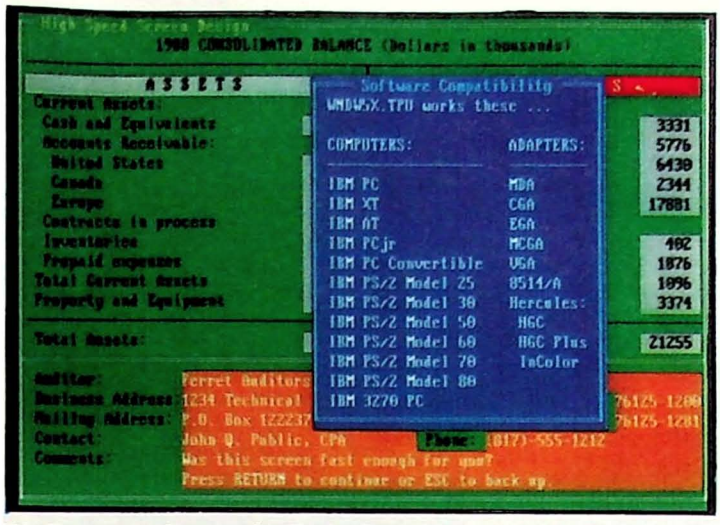
»Ganz klein mit Hut« wird eine riesige Datei mit dem richtigen Pack-Algorithmus. Eine faszinierende Sache. Die Hintergründe der Datenkompression beleuchtet ein Artikel auf **Seite 62**



Der Versionssprung von MS-DOS 3.3 zum MS-DOS 4.01 veränderte das Outfit dieses bekannten Betriebssystems gewaltig. Doch auch sein Befehlsumfang wurde kräftig überarbeitet und erweitert. **Seite 33**



Die PC-Hardware ist vielen ein Brief mit sieben Siegeln. Aber beim Einbau Ihrer Festplatte beispielsweise kommen Sie um ein wenig Hardware-Wissen nicht herum. Ein Beitrag zeigt diesen Vorgang von A bis Z. **Seite 48**



Die Programmiersprache Turbo Pascal hat eine glänzende Karriere hinter sich. Selbst die Shareware-Szene hat sich dieser Sprache mit interessanten Utilities intensiv angenommen.

Seite 10

Hercules-Grafiken in Basic programmieren? Dazu schüttelt jeder Basic-Erfahrene skeptisch den Kopf. Aber es funktioniert ausgezeichnet, was unser Listing bravourös beweist.

Seite 84

Die Tips & Tricks gehören zur DOS Extra wie zum Auto die Räder. Dieses Mal haben wir eine Menge Tips aus den Hotlines bekannter Software-Häuser für Sie zusammengetragen.

Seite 64

Tips & Tricks

Tricks aus der Software-Hotline 64

Tips & Tricks zu dBase III Plus 74

Fragen und Antworten zu Quick Basic 78

Utilities

Hercules-Grafik in Basic programmiert 84
– *für Quick Basic, Turbo Basic und SBasic*

SBasic unter der Lupe 98

Verbesserungen zu SBasic 102

Der Bilderdieb 108
– *Snapshot Version 2.7*

Anwendung

Kopieren mit 115200 Bit pro Sekunde 124

Nützlicher Helfer im Hintergrund 133
– *TSR-Datei-Manager*

SAM III, Version 1.1 143

Der imaginäre Drucker 156
– *Trickreiche Dateiumleitung*

Rubriken

Editorial 3
– *Die richtige Mischung macht's*

SCheck 2.02 80
– *Checksummer für sichere Listing-Eingabe*

Impressum 162

Der Weg zur Unabhängigkeit

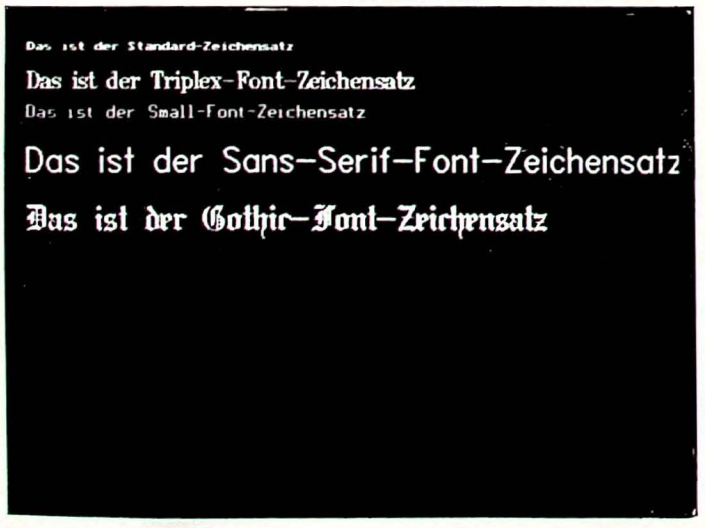
BGI-Treiber – Schnittstelle zur Grafikkarte

Wer bislang Grafikprogramme schrieb, tat dies für eine bestimmte Grafikkarte. Sollten dagegen unterschiedliche Standards berücksichtigt werden, mußten komplexe Installationsroutinen die Anpassung des Programms an die jeweilige Hardware sicherstellen. Ein zukunftssicheres und erweiterungsfähiges Grafikkonzept schafft hier mehr Unabhängigkeit.

Wenn Sie Ihren Computer mit einer Hercules-Grafikkarte ausgerüstet und einige Grafikroutinen dafür programmiert haben, wird dieses Programm sehr wahrscheinlich nicht auf einem Rechner mit EGA-Karte funktionieren. Da es für den PC inzwischen mehr als sechs verschiedene Grafikkarten gibt, müßten Sie für jede Karte ein eigenes Programm anfertigen.
Um dem Programmierer die Arbeit zu erleichtern, hat Borland daher für die Programmiersprachen Turbo Pascal und Turbo C ein sehr flexibles Konzept eingeführt – die BGI-Treiber (»Borland Graphics Interfaces«). Diese Dateien sind an der Dateierweiterung »bgi« auf der Diskette zu erkennen. Im Dateinamen steckt zusätzlich die Information, für welche Grafikkarte der jeweilige Treiber geeignet ist. In Ergänzung dazu liefert Borland einige Schriftarten (»Fonts«) mit, die eine ansprechende Gestaltung von Grafikprogrammen erlauben.
Eine Ausnahme bei den Fonts bildet der Standardzeichensatz. Er liegt nicht in Form einer Datei vor, sondern wird als Bitmuster direkt erzeugt. Im Gegensatz dazu sind die ladbaren Schriftarten sogenannte Vektor-Fonts. Das bedeutet, daß nur die Eckpunkte und charakteristischen Stellen der einzelnen Buchstaben definiert sind.
Die jeweiligen, auf dem Bildschirm erscheinenden Bildpunkte errechnet das Programm während der Ausgabe. Dadurch nimmt die Auflösung der Zeichen auch bei großen Buchstaben nicht ab.
Bei einem BGI-Treiber handelt es sich um eine Datei, die sämtliche Assemblerroutinen für die Ansteuerung einer bestimmten Grafikkarte enthält. Für jeden Kartentyp, den Sie verwenden wollen, muß ein Treiber existieren. Nur so läßt sich die jeweilige Grafikkarte ansprechen. Die Grafikbibliothek beziehungsweise der Compiler beinhaltet also keinerlei Code zur Ansteuerung der Grafikkarte.
Dieses auf den ersten Blick etwas umständlich erscheinende Konzept bietet Ihnen einige Vorteile: Zum einen ist es relativ einfach, ungewöhnliche Grafikformate oder neue Grafikkarten in bestehende Grafikprogramme einzubinden. Man schreibt »einfach« einen entsprechenden Treiber, und die Karte tut ihren Dienst. Natürlich ist diese Arbeit nichts für Anfänger, aber sie erfordert wesentlich weniger Aufwand, als eine Änderung des Compilers notwendig machen würde. Außerdem lassen sich die Erweiterungen den Besitzern der Grafikkarten leichter zugänglich machen. Es gibt auch schon einige Hersteller von nicht standardisierten Karten, die einen entsprechenden BGI-Treiber mitliefern.
Durch die Tatsache, daß die benötigten Programmroutinen in externen Dateien vorliegen, halten Sie Ihre Grafikprogramme so klein wie möglich. Wenn Sie beispielsweise eine EGA- oder VGA-Karte voraussetzen, so muß das fertige Programm nicht den Assemblercode für die restlichen Karten enthalten.
Ähnlich sieht es mit den Schriften aus, die zum Grafikpaket gehören. Diese Dateien weisen die Dateierweiterung »chr« auf, wobei der Name wiederum den Schrifttyp anzeigt. Neben dem platzsparenden Effekt ergibt sich hier der Vorteil, mit einem geeigneten Editor diese Schriften verändern oder an eigene Anforderungen anpassen zu können. Selbst das Erzeugen von gänzlich neuen Zeichensätzen stellt kein Problem dar.

Anpassungen an neue Grafikkarten lassen sich mit geringem Aufwand durchführen

Der dazu passende Editor und weitere Zeichensätze sowie einige verbesserte Treiber sind in vielen Mailboxen zu finden. Auf Anfrage hilft Borland registrierten Benutzern weiter. In einigen Software-Paketen von Borland (zum Beispiel dem relationalen Datenbanksystem »Paradox« oder auch auf der Paradox-Demodiskette) finden Sie außerdem neue Zeichensatzdateien, die Sie natürlich auch verwenden können. Dadurch erweitern Sie Ihre Auswahl an Schriften.
Sämtliche Treiber- und Schriftartendateien sind für alle Compiler von Borland die gleichen. Damit bereitet die Übertragung eines neuen Treibers auf eine andere Programmiersprache keine Schwierigkeiten mehr.
Anhand von zwei Beispielen sehen Sie, wie Sie diese Grafikunterstützung in Ihren eigenen Programmen verwenden können. Grundsätzlich gibt es zwei Alternativen, die Treiber einzusetzen. Die platzsparende Methode ist die, den jeweils benötigten Treiber während des Programmlaufs nachzuladen. Diese Aufgabe übernimmt der Compiler beziehungswei-



In den Programmcode eingebundene Zeichensätze benötigen keine zeitaufwendigen Diskettenzugriffe

se das erzeugte Programm automatisch.
Sollten Sie noch zusätzliche Schriftarten benötigen, so lassen sich diese – genau wie die Treiberdateien – von der Festplatte oder einer Diskette nachladen. Dadurch wird das übersetzte Programm auf Ihrem Speichermedium erheblich kleiner, und für die nachträgliche Anpassung an neue Grafikkarten ist keine Neuübersetzung (Compilieren) Ihres Quelltextes notwendig.
Allerdings müssen während der Programmlaufzeit die benötigten Treiber auf der Festplatte verfügbar sein. Diese Art der Verwendung ist im Programmbeispiel »grafik1.pas« (Listing 1) dargestellt.
Durch die Anweisung »uses Graph;« teilen Sie dem Compiler zunächst mit, daß Sie die Grafikbibliothek verwenden möchten. Diese Anweisung ist bei jedem Programm notwendig, das Grafikbefehle enthält. Für das Einschalten des Grafikmodus ist die Prozedur »InitGraph« zuständig. Ihr werden über drei Parameter die wichtigsten Einstellungen mitgeteilt. Über die Variable »Grafiktreiber« legen Sie fest, welcher BGI-Treiber geladen werden soll. Die vordefinierte Konstante »detect« weist das Programm an, selbst festzustellen, welche Karte eingebaut ist, und anschließend den entsprechenden Treiber zu laden.

Mit nachladbaren Treibern lassen sich Grafikprogramme sehr platzsparend schreiben

Diese automatische Erkennung funktioniert allerdings nur bei Standardkarten, das heißt bei den Karten, für die Borland die BGI-Dateien mitliefert. Andere Treiber müssen Sie zuerst über die Anweisung »RegisterBGIDriver« einbinden und den Grafikmodus dann gezielt setzen.
Der zweite Parameter legt den zu verwendenden Grafikmodus fest. Weil zum Beispiel die VGA-Karte auch eine EGA-Karte emulieren kann, sind Sie somit in der Lage, auch diesen Modus zu nutzen. Ebenso lassen sich verschiedene Bildschirmseiten auswählen. Im Beispiel wird immer die höchstmögliche Auflösung gewählt. Dies ist bei der automatischen Erkennung immer der Fall.
Zuletzt müssen Sie noch angeben, in welchem Verzeichnis beziehungsweise auf welchem Laufwerk die benötigten Treiber- und Font-Dateien zu finden sind. Durch Angabe eines Leer-Strings erreichen Sie, daß der Compiler diese Dateien im aktuellen Verzeichnis sucht.
Nach diesen Vorbereitungen ist der Grafikmodus eingeschaltet. Um eventuell auftretende Fehler abzufangen, gibt es die Funktion »GraphResult«. Sie liefert beim Aufruf den Fehlercode der letzen Grafikoperation. Sollte das compilierte Programm die eingebaute Grafikkarte nicht erkennen oder kann es die benötigten Treiber nicht finden, so wird das Beispielprogramm mit einer Fehlermeldung abgebrochen. Die Konstante »GrOK« entspricht dem Wert »0« und ist in der Unit »Graph« definiert.
Um die Grafik zu testen und die Verwendung der Schriften zu zeigen, stellt das Beispielprogramm »grafik1.pas« alle mitgelieferten Zeichensätze dar. Die jeweilige Schriftart beeinflussen Sie mit der Prozedur »SetTextStyle«, die wiederum drei Parameter erwartet. Als erstes müssen Sie die Zeichensatznummer angeben. Der Compiler verwaltet insgesamt zehn Zeichensätze. Solange Sie nur mit den vier im Lieferumfang enthaltenen Zeichensätzen arbeiten, können Sie dafür die vordefinierten Konstanten verwenden. Wie Sie andere Fonts einbinden müssen, zeigt Ihnen das zweite Beispiel.
Der zweite Parameter legt die Ausgaberichtung fest, wobei Sie die Wahl haben zwischen waagerecht (von links nach rechts) und senkrecht (von unten nach oben). Zuletzt müssen Sie noch die Größe des auszugebenden Textes angeben.
Über die Funktion »GraphResult« können Sie auch in diesem Fall das erfolgreiche Laden der Zeichensatzdateien kontrollieren. Im Beispiel erfolgt keine explizite Abfrage. Sollte eine Font-Datei nicht verfügbar oder nicht lesbar sein, verwendet der Compiler automatisch den Standardzeichensatz.
Die letzte Anweisung »CloseGraph« schaltet den Bildschirm wieder in den Textmodus zurück und schließt alle Grafikdateien ordnungsgemäß. Diese Prozedur müssen Sie immer am Ende einer Grafikausgabe anwenden.
Beim Ablauf des ersten Beispiels (»grafik1.pas«) werden Sie feststellen, daß der Computer ständig auf die Diskette oder Festplatte zugreift. Dies kennzeichnet die Ladevorgänge des Compilers für die Grafikroutinen. Die jeweiligen Treiber und Schriftarten werden nämlich immer dann in den Arbeitsspeicher geladen, wenn das Programm sie anfordert. Falls Sie keine Festplatte besitzen oder das fertige Programm häufiger von Diskette starten müssen, machen sich diese Ladezeiten deutlich bemerkbar. Turbo Pascal bietet Ihnen aber Abhilfe für dieses Problem.
Im Gegensatz zum Laden von Treiber- und Font-Dateien während des Programmlaufs lassen sich alle benötigten Grafiktreiber sowie die Schriftarten bereits während des Compilierens in das Programm einbinden. So erhalten Sie ein einziges Programm, und die BGI-Treiber und Zeichensatzdateien müssen zum Einsatz des Programms nicht mehr verfügbar sein (Bild). Professionellen Programmierern ist diese Möglichkeit anzuraten, denn eine einzige Programmdatei ist leichter zu handhaben als eine ganze Reihe von Dateien.

Das Einbinden aller Routinen erspart das Nachladen von Dateien

Leider kann aber dieses eine Programm unter Umständen sehr groß geraten, weil alle benötigten Daten im Programmcode enthalten sein müssen. Soll Ihre Grafikanwendung auf allen üblichen Konfigurationen laufen und haben Sie vielleicht sogar zehn Schriftarten vorgesehen, dann wird das Programm allein durch diese Dateien 170 KByte groß. Je nach Umfang des Quelltextes überschreiten Sie schnell die 256-KByte-Grenze, die als Vorgabe hinsichtlich des Arbeitsspeichers mancher PCs durchaus noch besteht. Diese Variante ist daher besonders für die Fälle zu empfehlen, in denen die zu verwendende Grafikkarte feststeht und nur wenige Schriftarten zum Einsatz kommen.
Im zweiten Beispielprogramm »grafik2.pas« (Listing 2) werden ein Treiber und vier Fonts eingebunden. Diese Arbeit übernimmt bei der Programmübersetzung der sogenannte Linker, dessen Existenz Sie normalerweise nicht bemerken. Da in diesem Fall mehrere Dateien zu einem einzigen Programm zusammengefügt werden sollen, müssen alle Dateien in einem bestimmten Format (dem sogenannten Object-Format) vorliegen. Der Compiler erzeugt dieses Format automatisch aus dem Quelltext – im Falle der BGI-Treiber und Zeichensätze allerdings hat der Programmierer diese Arbeit zu erledigen. Zu diesem Zweck enthält das Compiler-Programmpaket das Hilfsprogramm »binobj.exe«. Das Beispiel verwendet den kombinierten EGA-VGA-Treiber und die vier Zeichensätze. Diese Dateien sind daher zu konvertieren. (Um das Beispiel mit Erfolg nachvollziehen zu können, müssen Sie natürlich den für Ihre Grafikkarte passenden Treiber verwenden.)
Für den BGI-Treiber ist folgende Zeile der MS-DOS-Ebene einzugeben:

```
binobj egavga.bgi egavga.obj egavgadri-
verproc
```

Dadurch weisen Sie das Programm »Binobj« an, den Treiber »egavga.bgi« in das benötigte Format (»egavga.obj«) zu konvertieren. Zusätzlich erhält die Object-Datei den für den Linker lesbaren Namen »egavgadriverproc«. Ebenso verfahren Sie mit den Font-Dateien. Die jeweils verwendeten Namen entnehmen Sie dabei dem Listing 2.
Im Quelltext werden diese Dateien dann als externe Prozeduren deklariert. Direkt nach jeder Deklaration muß der Compiler-Befehl »{$L <Dateiname>}« stehen. Der Linker erhält dadurch die Anweisung, die entsprechende Datei in den Programmcode einzubinden.
Damit sind zwar die benötigten Daten bereits im Programmcode enthalten, der Compiler muß aber auch wissen, an welcher Adresse er die entsprechenden Routinen findet. Dies teilen Sie ihm für die BGI-Treiber durch die Funktion »RegisterBGIDriver« mit. Als Parameter ist ein Zeiger auf die Adresse der Routine notwendig. Turbo Pascal ermittelt diese Adresse selbständig, wenn Sie dem Namen der Programmroutine ein »@« voranstellen. Die Zeichensätze melden Sie auf die gleiche Weise mit der Funktion »RegisterBGIFont« an.

Am Beispiel des Triplex-Fonts sehen Sie, wie sich zusätzliche Zeichensätze einbinden lassen. Die Funktion gibt eine Integer-Zahl zurück, die Sie anstelle einer vordefinierten Konstanten verwenden. Unabhängig davon, ob der Zeichensatz geladen werden muß oder – wie hier – schon im Programmcode enthalten ist: Auf alle Fälle müssen Sie ihn vor der ersten Verwendung registrieren.

Der Rest des Programms gleicht dem ersten Beispiel. Die Pfadangabe als Parameter der Prozedur »InitGraph« ist allerdings hinfällig, weil keine Daten nachgeladen werden müssen. Obwohl der Inhalt dieses Parameters hier nicht von Bedeutung ist, muß er der Vollständigkeit halber dennoch angegeben werden.

Die Parameter »Grafiktreiber« und »Grafikmodus« müssen Sie natürlich noch auf den gewünschten Modus setzen. Im Beispiel ist das der VGA-Modus mit einer Auflösung von 640x480 Bildpunkten. Eine automatische Erkennung ist nur dann sinnvoll, wenn Sie alle BGI-Treiber in Ihr Programm einbinden.

Sofern Sie versuchen, Zeichensätze zu verwenden, die eine Länge von 15360 Byte (3C00hex) überschreiten, werden Sie feststellen, daß der Compiler diese nicht mehr erkennt. Der Grund dafür liegt in der Unit »graph.tpu«. Diese Unit enthält eine Abfrage, ob der Zeichensatz länger als eben diese 15360 Byte ist. Um das Problem zu beseitigen, müssen Sie diese Unit »patchen«, das heißt, die beiden Bytes ändern, die den Grenzwert dieser Abfrage festlegen. Dazu schreiben Sie sich eine ASCII-Datei mit folgendem Inhalt (gültig für die Unit »graph.tpu« von Turbo Pascal 5.5):

```
n graph.tpu
l
e 2ccd 5c
w
q
[Enter]
```

Nachdem Sie sichergestellt haben, daß sich der MS-DOS-Debugger »debug«, die Turbo-Pascal-Unit »graph.tpu« und die ASCII-Datei »patch.dat« im gleichen Verzeichnis befinden, geben Sie in der MS-DOS-Befehlsebene folgende Kommandozeile ein:

```
debug < patch.dat
```

Weil Sie den Grenzwert für die Zeichensatzlänge damit auf 23552 Byte erhöht haben, sollten nun alle Schriftarten einwandfrei zu verwenden sein. (Guido Weckwerth/zi)

Name	grafik1.pas
Sprache	Turbo Pascal ab Version 5.Ø
Grafikkarte	VGA, EGA
Besonderheiten	keine

```
[885]    1: { Programmname: GRAFIK1.PAS  }
[V94]    2: { (C)  Everts & Hagedorn     }
[ZB4]    3: {      DOS International     }
[ØØØ]    4:
[MS2]    5: program GRAFIK1;
[ØØØ]    6:
[CØ2]    7: uses Graph;
[9U5]    8: { Bibliothek der Grafikroutinen }
[ØØØ]    9:
[5B3]   1Ø: var  Grafiktreiber,
[362]   11:      Grafikmodus,
[FN3]   12:      Fehlercode   : integer;
[ØØØ]   13:
[MUØ]   14: begin
[GØ4]   15:   Grafiktreiber := detect;
[7N4]   16: { automatische Erkennung }
[CT6]   17:   InitGraph(Grafiktreiber,Grafikmodus,'');
[SF7]   18: { sucht Treiber im aktuellen Verzeichnis }
[ØØØ]   19:
[VA4]   2Ø:   Fehlercode := GraphResult;
[725]   21:   if Fehlercode <> grOK then begin
[BØ6]   22:     writeln (' Fehler beim Erkennen der '+
[BH2]   23:              'Grafikkarte ');
[Z76]   24:     writeln (' Programm abgebrochen ... ');
[K71]   25:     Halt (1);
[ZNØ]   26:   end;
[ØØØ]   27:
[5S6]   28:   SetTextStyle (DefaultFont, HorizDir, 1);
[PE6]   29:   OutTextXY (Ø,1,'Das ist der Standard-'+
[1K2]   3Ø:              'Zeichensatz');
[5V6]   31:   SetTextStyle (TriplexFont, HorizDir, 2);
[D86]   32:   OutTextXY (Ø,2Ø,'Das ist der Triplex-'+
[V93]   33:              'Font-Zeichensatz');
[7H6]   34:   SetTextStyle (SmallFont, HorizDir, 6);
[TT5]   35:   OutTextXY (Ø,5Ø,'Das ist der Small-'+
[V93]   36:              'Font-Zeichensatz');
[M37]   37:   SetTextStyle(SansSerifFont, HorizDir, 4);
[6L5]   38:   OutTextXY (Ø,8Ø,'Das ist der Sans-'+
[MF4]   39:              'Serif-Font-Zeichensatz');
[YR6]   4Ø:   SetTextStyle (GothicFont, HorizDir, 4);
[DØ6]   41:   OutTextXY (Ø,13Ø,'Das ist der Gothic-'+
[V93]   42:              'Font-Zeichensatz');
[991]   43:   readln;
[522]   44:   CloseGraph;
[ØØØ]   45:
[KNØ]   46: end.
```

Listing 1. Das Programm »grafik1.pas« demonstriert die Verwendung von Treiber- und Font-Dateien

Name	grafik2.pas
Sprache	Turbo Pascal 5.Ø
Grafikkarte	VGA (EGA)
Besonderheiten	keine

```
[C85]    1: { Programmname: GRAFIK2.PAS  }
[V94]    2: { (C)  Everts & Hagedorn     }
[ZB4]    3: {      DOS International     }
[ØØØ]    4:
[SS2]    5: program GRAFIK2;
[ØØØ]    6:
[1C6]    7: {=======================================
[4V5]    8:   Anweisungen zum Vorbereiten der
[KH5]    9:   Zeichensätze und Grafiktreiber
[3F3]   1Ø:   ------------------------------
[ØØØ]   11:
[9N5]   12:   Konvertieren des Grafiktreibers
[ØØØ]   13:
[KK5]   14:   (Verwenden Sie hier den für Ihre
[4E5]   15:    Grafikkarte bestimmten Treiber!)
[ØØØ]   16:
```

```
[3H7]    17: BINOBJ EGAVGA.BGI EGAVGA.OBJ EgaVgaDriverPr
             oc
[ØØØ]    18:
[DB5]    19:   Konvertieren der Zeichensätze
[ØØØ]    2Ø:
[KW6]    21: BINOBJ GOTH.CHR GOTH.OBJ GothicFontProc
[2A7]    22: BINOBJ SANS.CHR SANS.OBJ SansSerifFontProc
[UN6]    23: BINOBJ LITT.CHR LITT.OBJ SmallFontProc
[L27]    24: BINOBJ TRIP.CHR TRIP.OBJ TriplexFontProc
[ØØØ]    25:
[TR6]    26: ========================================}
[ØØØ]    27:
[CØ2]    28: uses Graph;
[9U5]    29: { Bibliothek der Grafikroutinen }
[ØØØ]    3Ø:
[5B3]    31: var  Grafiktreiber,
[362]    32:      Grafikmodus,
[9X1]    33:      Fehlercode,
[GZ3]    34:      Zeichensatz  : integer;
[ØØØ]    35:
[5H6]    36: { Einbinden der vier Zeichensätze }
[ØØØ]    37:
[886]    38: procedure GothicFontProc; external;
[RG2]    39: {$L GOTH.OBJ }
[GW6]    4Ø: procedure SansSerifFontProc; external;
[SF2]    41: {$L SANS.OBJ }
[N36]    42: procedure SmallFontProc; external;
[VG2]    43: {$L LITT.OBJ }
[7D6]    44: procedure TriplexFontProc; external;
[PG2]    45: {$L TRIP.OBJ }
[ØØØ]    46:
[VY5]    47: { Einbinden des EGA-VGA-Treibers }
[ØØØ]    48:
[FH6]    49: procedure EgaVgaDriverProc; external;
[2H2]    5Ø: {$L EGAVGA.OBJ }
[ØØØ]    51:
[MUØ]    52: begin
[1U6]    53: { Registrierung des EGA-VGA - Treibers }
[2H3]    54:   if RegisterBGIdriver(
[9K3]    55:      @EGAVGADriverProc) < Ø
[872]    56:   then  Halt (1);
[ØØØ]    57:
[EM6]    58: { Registrierung aller Zeichensätze }
[RG6]    59:   if RegisterBGIfont(@GothicFontProc) < Ø
[872]    6Ø:   then  Halt (1);
[7W6]    61:   if RegisterBGIfont(@SansSerifFontProc)<Ø
[872]    62:   then  Halt (1);
[276]    63:   if RegisterBGIfont(@SmallFontProc) < Ø
[872]    64:   then  Halt (1);
[ØØØ]    65:
[PD5]    66:   Zeichensatz := RegisterBGIfont(
[U83]    67:                        @TriplexFontProc);
[9Ø5]    68:   if Zeichensatz < Ø then  Halt (1);
[ØØØ]    69:
[LE3]    7Ø:   Grafiktreiber := VGA;
[KV3]    71:   Grafikmodus   := VGAhi;
[UV5]    72: { VGA: 64Øx48Ø Bildpunkte, 16 Farben }
[ØØØ]    73:
[3X6]    74:   InitGraph (Grafiktreiber, Grafikmodus, ''
             );
[SF7]    75: { sucht Treiber im aktuellen Verzeichnis }
[ØØØ]    76:
[VA4]    77:   Fehlercode := GraphResult;
[725]    78:   if Fehlercode <> grOK then begin
[BØ6]    79:     writeln (' Fehler beim Erkennen der '+
[BH2]    8Ø:              'Grafikkarte ');
[Z76]    81:     writeln (' Programm abgebrochen ... ');
[K71]    82:     Halt (1);
[ZNØ]    83:   end;
[ØØØ]    84:
[5S6]    85:   SetTextStyle (DefaultFont, HorizDir, 1);
[PE6]    86:   OutTextXY (Ø,1,'Das ist der Standard-'+
[1K2]    87:              'Zeichensatz');
[6T6]    88:   SetTextStyle (Zeichensatz, HorizDir, 2);
[D86]    89:   OutTextXY (Ø,2Ø,'Das ist der Triplex-'+
[V93]    9Ø:              'Font-Zeichensatz');
[7H6]    91:   SetTextStyle (SmallFont, HorizDir, 6);
[TT5]    92:   OutTextXY (Ø,5Ø,'Das ist der Small-'+
[V93]    93:              'Font-Zeichensatz');
[M37]    94:   SetTextStyle(SansSerifFont, HorizDir, 4);
[6L5]    95:   OutTextXY (Ø,8Ø,'Das ist der Sans-'+
[MF4]    96:              'Serif-Font-Zeichensatz');
[YR6]    97:   SetTextStyle (GothicFont, HorizDir, 4);
[DQ6]    98:   OutTextXY (Ø,13Ø,'Das ist der Gothic-'+
[V93]    99:              'Font-Zeichensatz');
[991]   1ØØ:   readln;
[522]   1Ø1:   CloseGraph;
[ØØØ]   1Ø2:
[KNØ]   1Ø3: end.
```

Listing 2. »grafik2.pas« bindet die benötigten Treiber in den Programmcode ein

3 überzeugende Pluspunkte der DOS-TEST-Konzeption:

Vergleichbare Testberichte mit einheitlichen Kriterien helfen Ihnen bei der Kaufentscheidung.

Knallharte Information über interessante Themen, verständlich und locker mit viel Lesespaß.

Durch Tests im praktischen Einsatz wissen Sie schon heute, welche Hard- und Software morgen aktuell ist.

Jetzt bei Ihrem Zeitschriftenhändler

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Shareware-Tools für Turbo Pascal

Wer nur für einen bestimmten Bereich Programme entwickelt, erlernt dazu vorwiegend problemorientierte Sprachen. Ist das Entwicklungsgebiet aber breit gefächert, hat man nur die Wahl, entweder mehrere problemorientierte Sprachen zu beherrschen oder eben in Pascal oder C zu programmieren. Dann stößt man allerdings auf das Problem, sich bestimmte Funktionen selbst schreiben zu müssen, die beispielsweise Cobol schon von vornherein enthält. Wer diesen Aufwand nicht auf sich nehmen will, ist mit Toolboxen gut bedient.

Haben Sie schon einmal versucht, ein Pascal-Programm zu schreiben, das Eingaben von der Tastatur für eine bestimmte Feldlänge zuläßt und darauf achtet, daß nur zulässige Zeichen in die Eingabe kommen? Auch wenn dies noch eine ziemlich leichte Übung ist, bedeutet sie doch Programmier- und damit Zeitaufwand. Besser greift man da zu Toolboxen, die solche und viele andere Funktionen bereitstellen. Davon gibt es eine ganze Reihe auch im Shareware-Markt.

Während sich »WNDW 5.0« ausschließlich mit Fenstern beschäftigt und »Pull« mit Menüs, kann man das »Turbo-Pascal-Enhancement-Toolkit« und »Boosters 4.0« eher als Toolbox für alles bezeichnen. Alle Toolboxen widmen sich eher der »seriösen« Programmierung und sparen daher Grafikroutinen weitgehend aus.

Die **Boosters-Toolbox** ist schon seit Turbo Pascal 3.0 bekannt – so finden Sie in ihr Funktionen, die Turbo Pascal 4.0 bereits unterstützt: »Rename«, »Erase«, »GetFTime«, »GetFAttr« etc. Diese Routinen ab Turbo Pascal 4.0 arbeiten allerdings alle mit einer Dateivariablen, während die entsprechenden Funktionen (und Prozeduren) von Boosters direkt mit dem Dateinamen arbeiten, ohne daß zuvor Dateivariablen zuzuweisen sind. Für die Bearbeitung von Dateien stellt die Toolbox sonst nur noch »BoDir« zur Verfügung. Damit präsentieren Sie dem Benutzer durch einen Prozeduraufruf das Inhaltsverzeichnis in einem Fenster wählbarer Größe und Umrahmung, woraus der Anwender dann Dateien in der Art auswählt, wie man es von der Pascal-Oberfläche gewohnt ist.

Die Bildschirmfunktionen beschäftigen sich in erster Linie mit dem Bildschirm als Speicherbereich. So kann man ihn auf den Heap kopieren oder von dort an den Monitor schicken, selbst vom Massenspeicher laden und darauf speichern. Bilder lassen sich übrigens mit dem integrierten Bildschirmeditor entwerfen.

Eine einfache Box-Prozedur malt Kästen auf den Bildschirm, deren gegenüberliegende Seiten allerdings nur von gleicher Linienart sein können. Ausschnitte können Sie einfärben, löschen, kopieren und über den Bildschirm verschieben (Bild 1). Dabei füllen Sie den überfahrenen Platz mit Leerzeichen beliebigen Attributs oder rekonstruieren ihn wieder. Eine weitere Prozedur malt einen Rahmen, löscht ihn wieder, um dann zwei entsprechende Rahmenzeichen um den Bereich kreisen zu lassen wie Sputniks – wohl eher eine Spielerei. Einzelne Zeichen und beziehungsweise oder deren Attribute, sogar Strings lassen sich direkt vom Bildschirm lesen oder ändern. Besondere Beachtung verdient die Prozedur »FindStr«, die den Bildschirm nach einer Zeichenkette durchsucht und ihre Koordinaten zurückgibt. Fast alle dieser Funktionen können sich auch auf die im Heap befindlichen Bildschirminhalte beziehen. Bilder, die dort gespeichert sind, bringen Sie mit den unterschiedlichsten Bildübergängen zur Ansicht – die zwei Hälften des ersten Bildes schieben sich beispielsweise vertikal oder horizontal auseinander oder zusammen.

Turbo Pascal behandelt die Bildschirmein- und Bildschirmausgabe etwas stiefmütterlich. Das gleichen Sie mit den Prozeduren von Boosters wieder aus. Eine umfassende Menüfunktion fehlt zwar, mit »ChooseOption« wählen Sie aber aus einer bestimmten Anzahl von Elementen auf dem Bildschirm aus, die zuvor mit »Write« oder der Boosters-Prozedur »pdq« auf den Monitor gebracht wurden. »pdq« schreibt eine Zeichenkette in allen vier Richtungen mit beliebigem Attribut auf den Bildschirm. Das geschieht direkt über den Bildschirmspeicher – entsprechend schnell ist denn auch die Prozedur.

Der Input-Befehl verspricht formatierte Eingabe. In einer übergebenen String-Variablen steht, ob auf der Position im Feld jedes beliebige Zeichen, nur Ziffern, Ziffern und Leerzeichen oder nur Vorzeichen eingegeben werden dürfen. Steht ein Leer-

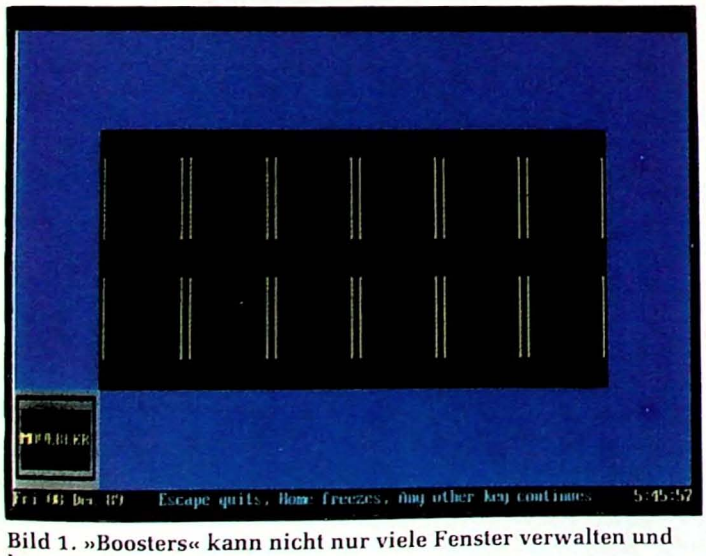


Bild 1. »Boosters« kann nicht nur viele Fenster verwalten und bewegen

Bild 2. »Turbo Enhancement Toolkit« zeigt in dem Demo seine Tools in Aktion

zeichen darin, überspringt »Input« die entsprechende Stelle und setzt die Eingabe dahinter fort. Dazwischen kann sich also noch Text befinden, der von der Eingaberoutine nicht beachtet wird. Die Form und Position des Cursors läßt sich ändern, oder Sie schicken ihn mit »DiffOne« an eine Position, wo ein Zeichen steht, das mit dem übergebenen nicht übereinstimmt.

Zu den String-Funktionen gibt es eigentlich nicht viel zu sagen: Alles, was man irgendwie brauchen kann, ist dabei – Vervielfältigen, Zählen, Suchen, Ausschneiden, Einfügen oder Löschen von Buchstaben, Wörtern und Bereichen. Zahlen vom Typ »LongInt« wandeln Sie in Strings oder hexadezimale in dezimale Werte um. Strings positionieren Sie innerhalb einer vorgegebenen Länge, umgeben von Zeichen freier Wahl, links, rechts oder zentriert. Überflüssige Zeichen (zum Beispiel Leerzeichen) schneiden Sie links oder rechts weg, oder Sie fügen beliebige Zeichen zwischen den Wörtern ein. Ganze Strings wandeln Sie in Großschrift beziehungsweise Kleinschrift um, oder Sie untersuchen, ob sie nur aus Zeichen bestehen, die in einer übergebenen String-Variablen stehen.

Für spezielle Anwendungen finden Sie die Prozedur »Calendar«, nach deren Aufruf Sie den Kalender eines bestimmten Monats und Jahrs vorgestellt bekommen.

Auch das **Enhancement Toolkit** ist eine Sammlung von Routinen »querbeet« (Bild 2). Der Schwerpunkt liegt dabei auf der Bildschirmein- und Bildschirmausgabe. Hier finden Sie Routinen, wie sie Turbo Pascal ab Version 4.0 schon zur Verfügung stellt, daneben aber auch alles, was zur Verwaltung von Fenstern und Bildschirmbereichen notwendig ist. So positioniert sich der Cursor entweder relativ zu Bereichen wie etwa Bildschirmausschnitten oder zum ganzen Bildschirm. Es beziehen sich viele Befehle explizit auf die Textspalten oder -zeilen des Bildschirms; die Zeilen- beziehungsweise Spaltenzahl ist dabei variabel.

Die Ausschnittumrahmung ist gegenüber derjenigen von Boosters besser gelungen. Sie können die Zeichen wählen, die für die horizontalen und vertikalen Linien zu verwenden sind. Auch jedes Eckenzeichen ist frei wählbar. Allerdings gilt auch hier: Die jeweils gegenüberliegenden Seiten setzen sich aus den gleichen Zeichen zusammen. Sonstige Ausschnittfunktionen gleichen denen von Boosters. Sie können füllen, einfärben und kopieren, wobei Sie entweder Zeichen oder Attribute oder beides angeben. Allerdings lassen sich Ausschnitte oder ganze Bildschirme auch scrollen.

Der Befehl »EditSt« läßt das freie Editieren einer Zeichenkette auf dem Bildschirm zu. Sie können dabei sogar den Startpunkt des Cursors im Feld angeben. Alle Sondertasten (Cursor-Tasten, [Backspace], [Home], [End], [Esc] etc.) stellen jeweils diejenige Funktion bereit, die man auch bei den üblichen Editoren findet. Außerdem gibt die Routine am Ende des Feldes einen Warnton von wählbarer Frequenz und Dauer aus. Ist dies unerwünscht, so setzen Sie einen Autoskip-Parameter auf »true«, und die Routine endet mit der Eingabe des letzten Feldelements. Die Tastenvorgabe ist etwas eingeschränkt: Sie übergeben dazu eine Variable vom Typ »set«, in der alle Zeichen als Bytewert stehen, die eingegeben werden dürfen. Das Zeichen, mit dem der Benutzer den Editor wieder verläßt, ist ebenfalls definierbar.

Das Enhancement Toolkit bietet auch Funktionen an, die sich mit den Videomodi beschäftigen: Augenblickliche Zeilen- und Spaltenzahl, Seitennummer, Bildschirmmodus und Adreß-Segmente der Bildschirmseiten lassen sich so ermitteln. Alle weiteren Funktionen beschäftigen sich intensiv mit dem BIOS und MS-DOS. Die Tastaturroutinen lassen beispielsweise das Löschen und Abfragen des Tastaturpuffers und die Abfrage des Tastaturstatusbytes zu, das den Zustand der Shift-, Alt- und Ctrl-Tasten beinhaltet (auch denjenigen von Num-, Caps-, Scroll-Lock und Insert).

Das Enhancement Toolkit löst glänzend, was selbst Boosters noch fehlt: die Betreuung der Maus. Jedes Mausereignis wird in einer Warteschlange vermerkt und abgearbeitet, sobald Zeit dazu ist. Sie können wählen, welche Ereignisse (Mausbewegung und Mausklick) zu registrieren sind. Sie können die Warteschlange auch

DOS-Blitzlicht

Name: Boosters 4.0
Programmfunktion: Turbo-Pascal-Toolbox für String-Funktionen und vieles mehr
Preis: 34,80 Mark (3 Disketten)
Registrierung: 25 Dollar
Voraussetzungen: Turbo Pascal ab Version 4
Benutzerführung: gut: Dokumentation auch als deutsches Buch (Turbo Pascal Tools 1 für 19,80 Mark)
Info: Systhema Verlag, 8000 München 82

Positiv:
- starke String-Funktionen
- Generator für Bildschirmgestaltung
- deutsche Dokumentation als Buch erhältlich

Negativ:
- unübersichtliche Parameterstruktur, weil zu viele Befehle

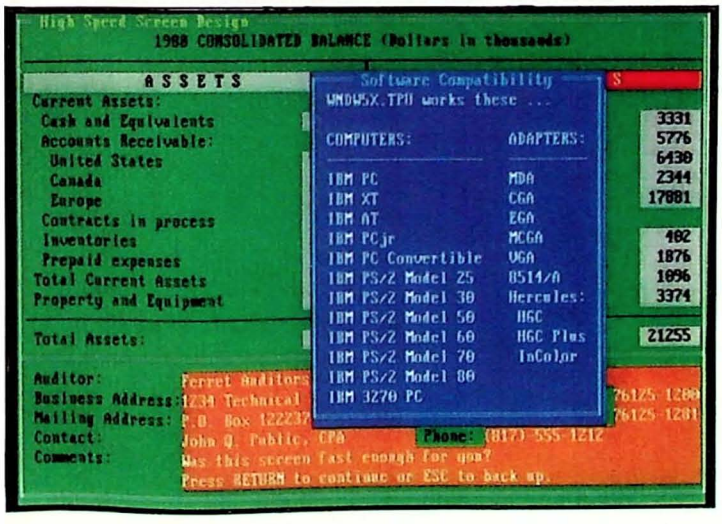


Bild 3. Das Demo von »WNDW« ist nur kurz, zeigt aber den Umgang mit Popup-Fenstern

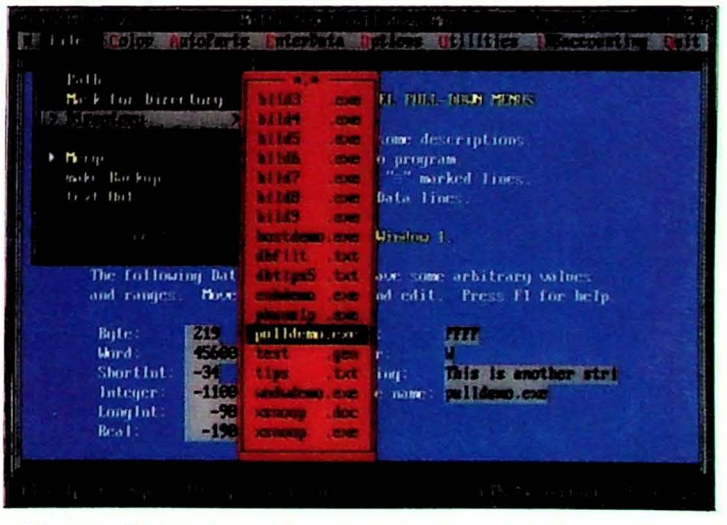


Bild 4. »Pull« in seinem Element: Das Demo zeigt Pulldown- und Popup-Fenster

löschen oder einzelne Elemente herausgreifen. In der Variablen »NEvents« steht, wie viele Ereignisse auf die Bearbeitung warten.
Der aktuelle Zustand einer Maustaste und die Position des Mauszeigers läßt sich explizit erfragen. Den Zeiger selbst können Sie definieren, auf dem Bildschirm erscheinen lassen, verstecken oder löschen. Auch die Form des Text-Cursors der Maus ist anzugeben. Mit »MSetXRange« und »MSetYRange« legen Sie die Bewegungsgrenzen des Mauszeigers fest und mit »MSetSpeed« die Übersetzungsgeschwindigkeit. Man kann sogar ein Mausereignis simulieren und so zum Beispiel den Zeiger wie von Geisterhand an eine gewünschte Position manövrieren.
Etwas eigentümlich muten die Dateifunktionen des Toolkit an. Zum Großteil handelt es sich dabei um Routinen, die Pascal in ähnlicher Weise schon bietet. Öffnen, Lesen, Schreiben und Schließen einer Datei ist ebenso enthalten wie das Positionieren des Dateizeigers. Das ganze lehnt sich nur näher an MS-DOS an und spricht bei einer Dateivariablen von einem »Handle«. Allerdings findet man auch Funktionen, um das aktuelle Laufwerk oder die Gesamtzahl logischer Laufwerke zu ermitteln. Neben »FindFirst« und »FindNext« (ab Turbo Pascal 4.0 schon enthalten) bekommen Sie die Funktionen »GetDTA« und »SetDTA« mitgeliefert, die bei Turbo Pascal leider fehlen.
Druckerfunktionen werden für wirklich professionelle Programmierung gefordert. Sie können zum Beispiel den Druckerstatus erfahren und das Gerät initialisieren. Die Prozedur zur Ausgabe einzelner Zeichen auf dem Drucker will nicht so ganz ihren Sinn erkennen lassen (»Write(LST,'?')«).

Vielleicht ist das Konzept der themenspezifischen Toolbox günstiger, weil Sie nur das erwerben, was Sie auch wirklich brauchen.
Dieser Ansicht waren wohl auch die Macher von »Pull« und **WNDW**. WNDW ist eine Toolbox, die Routinen zur Verwaltung mehrschichtiger Fenster bietet (Bild 3). Zunächst definieren Sie mit »SetWindowModes« den Fenstertyp. Ist der Zoom-Modus eingeschaltet, wächst das Fenster aus seiner Mitte heraus auf den Bildschirm, sobald es gezeigt werden soll. Sie können es auch schon versteckt erzeugen und erst nach »MakeWindow« mit »ShowWindow« zeigen. Schatten lassen sich links oder rechts darstellen. Hat das Fenster einen Cursor, ist sein Inhalt zu löschen? All das und mehr regelt dieser Befehl. Mit »InitWindow« übergeben Sie die Farbe des Fensters. »MakeWindow« vereinbart ein Fenster dann unter seinem Namen. Es hat auf Wunsch eine Titelzeile und einen von 13 vorgegebenen Rahmen. Ferner wählen Sie zwei frei definierbare Rahmenarten an. Von nun an verstecken, vergrößern, verkleinern, bewegen oder löschen Sie das Fenster nach Belieben.
Mit »AccessWindow« vereinbaren Sie ein Fenster für den Schreibzugriff und rücken es in den Vordergrund, falls es von anderen Fenstern überlagert sein sollte. Das Fenster zeigt auf Wunsch auch einen Ausschnitt aus einem virtuellen Bildschirm, dessen Maße nur durch den Speicher begrenzt sind und worin Sie das Fenster beliebig plazieren. Wahlweise schreiben Sie in das Fenster oder direkt auf den Bildschirm. Alle Elemente des Fensters lassen sich auch nachträglich ändern (Titel, Farbe, Rahmen, Cursor etc.). Innerhalb des Fensters verwenden Sie Pascal-übliche Kommandos, um einzelne Zeilen zu ändern (zum Beispiel »bis zum Zeilenende löschen«).
Die WNDW-Toolbox bietet in einer getrennten Unit Funktionen, die man generell und speziell für den Speicherzugriff braucht, beispielsweise schnelles Kopieren und Vergleichen. Alle Bildschirmroutinen sind extrem schnell, wenn man die mitgelieferte Unit »qwik« miteinbindet. Binden Sie »goof« ein, so zeigt ein Programm zur Laufzeit Fehler an, die mit dem Fenstersystem zusammenhängen.
Die Toolbox **Pull** besteht teilweise aus den Routinen von WNDW und Qwik, verwendet aber die Fensterverwaltung für ihre Menüfunktionen (Bild 4). Wie der Name schon vermuten läßt, handelt es sich dabei um Pulldown-Menüs. Nachdem Sie Position, Form, Farbe und andere Einstellungen vereinbart haben, übergeben Sie die Kontrolle an den Menüverwalter, und der liefert den angewählten Menüpunkt zurück.
Die Menüs sind mehrschichtig, überlagern sich also – Menüpunkte bringen wieder neue Menüs hervor. Innerhalb von Menüfenstern können editierbare Felder stehen oder geöffnet werden, wenn ein bestimmter Punkt angewählt wurde. Die darin eingegebene Tastenfolge steht in einer zuvor vereinbarten Variablen. Die Prozeduren und Funktionen, die dabei Verwendung finden, lassen sich alle auch direkt aufrufen. Fehlermeldungen oder auch andere Texte können Sie mit einem Aufruf in der Zeile »MessageLine« auf den Bildschirm bringen. Die Units aus WNDW 5.0 sind in Pull enthalten. Wer also Menüs benutzen möchte, aber eher zur Fensterverwaltung tendiert, sollte sich wohl eher gleich für Pull entscheiden.

(Dieter Bäumler/al)

DOS-Blitzlicht

Name: Turbo Pascal Enhancement Toolkit 3.1
Programmfunktion: Turbo-Pascal-Toolbox für Mausverwaltung und anderes
Preis: 16 Mark
Registrierung: 30 Dollar
Voraussetzungen: Turbo Pascal ab Version 3
Benutzerführung: befriedigend: englische Referenz mit auf der Diskette
Info: Kirschbaum Software, 8091 Emmering

Positiv:
- komplette Mausverwaltung
- umfangreicher Befehlssatz

Negativ:
- einige überflüssige Funktionen

DOS-Blitzlicht

Name: WNDW 5.0 und Qwik
Programmfunktion: Turbo-Pascal-Toolbox für Fenstertechnik
Preis: 12 Mark
Registrierung: je 25 Dollar
Voraussetzungen: Turbo Pascal ab Version 4
Benutzerführung: befriedigend: englische Referenz mit auf der Diskette
Info: Computer Solutions, 8018 Grafing

Positiv:
- sehr schnell
- umfangreicher Befehlsvorrat

Negativ:
- nur englische Dokumentation

DOS-Blitzlicht

Name: Pull
Programmfunktion: Turbo-Pascal-Toolbox für Pulldown-Menütechnik
Preis: 12 Mark
Registrierung: 25 Dollar, WNDW-Paket 50 Dollar
Voraussetzungen: Turbo Pascal ab Version 4
Benutzerführung: befriedigend: englische Referenz mit auf der Diskette
Info: Computer Solutions, 8018 Grafing

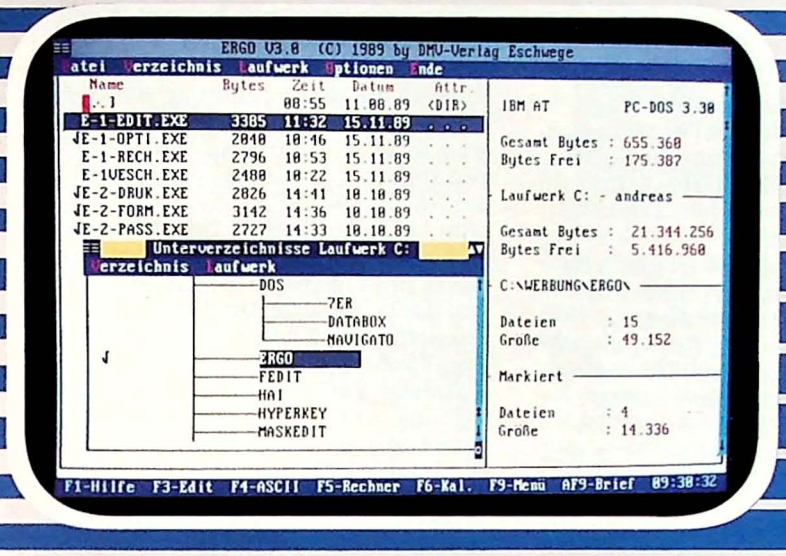
Positiv:
- enthält WNDW-Paket
- eigenständiger Menüverwalter
- situationsorientierte Hilfsfunktion programmierbar

Negativ:
- nur Pulldown-Menüs
- keine Mausverwaltung

DMV-Erfolgsprodukt '89
über **40.000mal**
verkauft

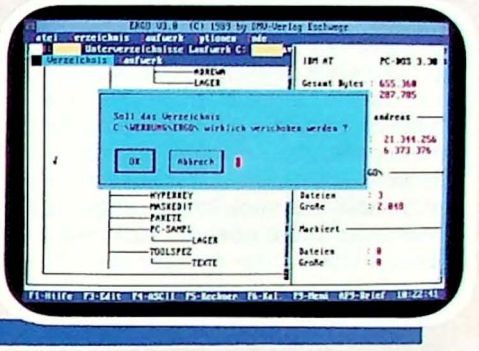
ERGO 3.0

Die DOS-Benutzeroberfläche, die einfach alles hat



Verzeichnisse verschieben

Ohne Ordnung ist Ihr Datenbestand nur die Hälfte wert. Daher können Sie bei ERGO nicht nur auf ein ausgefeiltes Dateihandling zurückgreifen – komplette Verzeichnisse lassen sich in einem Arbeitsgang verschieben und vor neugierigen Augen "verstecken".

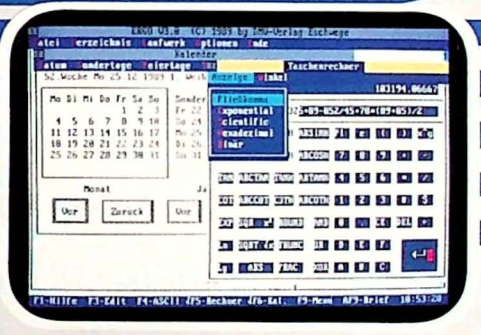


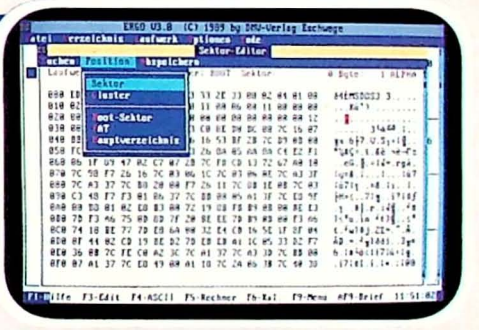
Taschenrechner und Kalender

Auch bei Terminplanungen können Sie sich auf ERGO verlassen: Ein Tastendruck, und der Kalender steht zur Verfügung. Der integrierte wissenschaftliche Taschenrechner ist durch die ERGO-Mausbedienung einfach zu benutzen und beherrscht auch Formelanalyse.

Sektor-Editor

Jetzt haben nicht mehr nur Profis den direkten Draht zur Datei. Mit ERGO suchen, ändern und speichern Sie einzelne Datei-Sektoren ab. Den komfortablen Byte-Editor können Sie selbstverständlich auch, wie das gesamte Programm ERGO, mit der Maus bedienen.





Dateien:	Kopieren, Löschen, Umbenennen, Verschieben, Attribute
Verzeichnisse:	Wechseln, Anlegen, Löschen, Umbenennen, Verstecken, Verschieben
Laufwerk:	Wechseln, Kopieren, Formatieren, Optimieren
Sektor-Editor:	Suchen, Weitersuchen, Position, Abspeichern, Wiederherstellen
Optionen:	Voreinstellungen, Paßwort, Zugriffsrechte, Druckercodes, Farben, Zeit/Datum setzen, TSR-Verwaltung, Systeminfo
DOS-Shell:	sofort in DOS – mit Exit zurück in ERGO
Hilfe-Funktion:	ASCII-Tabelle, Kalender, Online-Manual
Taschenrechner:	Fließkomma, exponential, scientific, hexadezimal, binär
Anwendermenü:	Programme lassen sich von diesem Menü aus starten
Editor:	Blockkommandos, Merkerfunktionen, Suchen und Ersetzen, Einrücken, Spaltenblock und Tabulator

Für IBM PC/XT/AT und 100% Kompatible mit mindestens 384 KByte RAM, PC-/MS-DOS ab Version 3.0, ältere Versionen mit Einschränkungen. Zur Verwendung der ERGO-Maussteuerung ist eine Microsoft-kompatible Maus erforderlich.

ERGO 3.0 DOS-Benutzeroberfläche — DM **199,–***

UPDATE auf ERGO 3.0 — DM **69,–***

DMV-Power-Pack
ERGO 3.0, ConText PRO, ConText ADREVA — DM **349,–***

Demonstrationsdiskette ERGO 3.0 — DM **5,–**

Dazu passend empfehlen wir:

HYPERKEY 4.0 Tastaturbelegung — DM **99,–***

* Unverbindliche Preisempfehlung. Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,– DM, bzw. für das Ausland 6,– DM Porto und Verpackung.

– Bitte benutzen Sie die Bestellkarte. –

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Units – Bibliotheken à la Pascal

Ab der Version 4.0 unterstützt Turbo Pascal das Unit-Konzept. Dem Pascal-Programmierer stehen damit vielfältige Bibliotheken zur Verfügung, die er – separat compiliert – nur in eigene Programme einzubinden braucht. Mit Routinen zur Datumsberechnung, komfortablen Tonerzeugung und (rekordverdächtigen) Zeichenausgabe auf dem Bildschirm bietet Ihnen dieser Beitrag Module an, die vielleicht den Grundstock Ihrer Unit-Sammlung bilden.

Einige recht praktische Datumsroutinen wurden zwar schon des öfteren beschrieben (beispielsweise in DOS 10'88), für die Darstellung vollständiger Kalender reichen sie jedoch nicht aus. Die Unit »calendar« (Listing 1) setzt dieser Unvollkommenheit ein Ende: Sie enthält alle notwendigen Funktionen für Kalenderberechnungen.

Im Sinne einer möglichst mühelosen Anwendung bietet sich ein Variablentyp »DateTyp« an, der dem Datenverbund

```
type DateTyp =
record
Year: word;
Month, Day: byte
end;
```

entspricht. Weil aber dieser Typ als Funktionsergebnis verwendet werden soll, ist eine derartige Definition in Turbo Pascal nicht zulässig (der Compiler akzeptiert nur einfache Datentypen, Strings und Zeiger als Funktionsergebnisse). Nun beansprucht »DateTyp« glücklicherweise nur 4 Byte Speicherplatz, und daher läßt sich seine Struktur vor dem Compiler gewissermaßen »geheimhalten«, wenn Sie zu Beginn einfach

```
type DateTyp = longint;
```

definieren.

Die innere Struktur legen Sie in dem (nicht öffentlichen) Verbundtyp »htyp« fest. Darum braucht sich jedoch ein Anwender der Unit nicht weiter zu kümmern. Die scheinbar vertauschte Anordnung der Variablen im Record »htyp« bewirkt, daß das Jahr »y« im höherwertigen Wort und der Tag »d« im niederwertigen Byte von »DateTyp« steht. Datumswerte lassen sich somit bequem vergleichen: Ein späteres Datum ergibt einen größeren Longint-Wert als ein früheres.

Wie Listing 1 zeigt, besteht »calendar.pas« aus zwölf Funktionen. Des weiteren sind die Konstantenfelder »WeekdayName« und »MonthName« vordefiniert, weil diese in den meisten Anwendungen benötigt werden.

Die Funktion »AddDate« verschiebt ein gegebenes Datum um »n« Tage auf ein späteres oder (bei einem negativen Parameter »n«) früheres Datum. Mit der Funktion »DiffDate« ermitteln Sie die Differenz zwischen zwei Datumswerten.

Die Verwendung von englischen Bezeichnern hat übrigens nichts mit »Snobismus« zu tun, sondern geschieht aus praktischen Erwägungen: Verwendet der Anwender für »seine« Variablen und Prozeduren konsequent deutsche Bezeichner und behält die englischen Ausdrücke in den Units bei, so treten niemals Namenskonflikte auf – ein Umstand, dem eine zur Anzahl der verwendeten Units proportional steigende Bedeutung zukommt.

Die Funktion »MakeDate« verwandelt einen String in einen Datumswert. Dies ist in Eingaberoutinen von großem Nutzen. Die Funktion ignoriert dabei alle Zeichen, die keine Ziffern darstellen. Enthält der String überhaupt keine Ziffer, dann liefert »MakeDate« das aktuelle Systemdatum Ihres PC zurück.

»MakeDate« – die Hilfsfunktion für eine völlig formative Datumseingabe

Sofern der String nur eine (ein- oder zweistellige) Zahl beinhaltet, wird diese als Tag interpretiert. Den fehlenden Monat und das Jahr ergänzt die Funktion durch die entsprechenden Angaben des Systemdatums. Kommen dagegen zwei derartige Zahlen im String vor, fügt »Make Date« nur noch die fehlende Jahresangabe (wiederum vom Systemdatum abgeleitet) an. Selbstverständlich addiert die Funktion zu zweistelligen Jahresangaben den Wert 1900. Unter der Voraussetzung, daß das Systemdatum Ihres PC das Jahr 1990 enthält, werden alle nachfolgenden Strings in das Datum »24.12.1990« konvertiert:

```
24121990
241290
Heute ist der 24. 12.
24/12
am 24sten 12ten 90 ist Weihnachten
24-12-1990
24 12 90
```

Mit der Funktion »DateError« lassen sich übergebene Datumswerte auf ihre Gültigkeit überprüfen. Das Demonstrationsprogramm »gebdat.pas« (Listing 2) zeigt Ihnen den Einsatz dieser Prüfroutine.

Für »selbstgestrickte« Kalender benötigen Sie die Datumswerte für Ostern und die beweglichen Feiertage (Pfingsten, Himmelfahrtstag). Nach dem – seit 1583 gültigen – Gregorianischen Kalender fällt der Ostersonntag auf jenen Sonntag, der dem ersten Vollmond nach Frühlingsbeginn folgt. Mondphasenberechnungen stellen an sich ein recht schwieriges Kapitel in der Astronomie dar. Man fand aber bereits im 16. Jahrhundert vereinfachte Regeln zur Bestimmung des Osterdatums. Diese wurden dann im Jahre 1800 von Carl Friedrich Gauß so verbessert, daß sie auch heute noch gut anwendbar sind. Auf diese Regeln stützt sich der Algorithmus, der in der Funktion »Easter« verwendet wird. »Easter« bildet das Datum des Ostersonntags mit dem Parameter »Jahr« im Bereich von 1700 bis 2199. Eine ungültige Jahresangabe (außerhalb dieses Bereichs) liefert den Datumswert »Null«.

Die Datumswerte der beweglichen Feste sind nun ganz einfach abzuleiten:

```
Ostersonntag := Easter(Jahr);
Karfreitag := AddDate(Ostersonntag, -2);
Himmelfahrtstag := AddDate(Ostersonntag,
39);
Pfingstsonntag := AddDate(Ostersonntag, 49);
```

Sämtliche Feiertagsbezeichner sind dabei als Typ »DateTyp« definiert. Das Programm »kalender.pas« (Listing 3) demonstriert, wie eine Kalenderübersicht aussehen könnte. Das Jahr ist als Programmparameter anzugeben; ohne Parameter wird das laufende Jahr angezeigt. Es wurden hier nur die im deutschen Sprachraum allgemein gültigen Feiertage aufgenommen. Das Feld »FesteFeiertage« ist daher, je nach Bedarf, »landesüblich« zu ergänzen. Das letzte Beispielprogramm zur Unit »Calendar« ist all jenen gewidmet, die sich zwar aus Horoskopen und vierblättrigen Kleeblättern nichts machen, es aber dennoch vermeiden, unter schrägstehenden Leitern hindurchzugehen, und schwarzen Katzen lieber ausweichen. Das kleine Programm »abergl.pas« (Listing 4) ermittelt bis zum Jahr 2010 alle Monate, in denen der 13. auf einen Freitag fällt. Aberglaube? Natürlich nicht – aber man weiß ja nie!

Wenn Sie als musisch veranlagter Mensch Ihre Programme mit etwas Musik ausschmücken möchten, dürfen Sie von Turbo Pascal – außer den Sound- und No-Sound-Anweisungen – keine Unterstützung erwarten. Die GW-Basic-Programmierer sind da weit besser bedient: Ihnen bietet die Play-Anweisung schon beachtliche »künstlerische« Entfaltungsmöglichkeiten bei der Wiedergabe beliebiger Melodien.

Mit Musik geht alles besser – die Music-Unit gibt den Ton an

Einen ähnlichen Gestaltungsspielraum können Sie aber auch in eigenen Turbo-Pascal-Programmen nutzen – sofern Sie die Unit »music.pas« (Listing 5) einbinden. Mit einer Pseudo-Notenschrift lassen sich auch hier ohne viel Aufwand ganze »Musikstücke« programmieren. Sie werden der abspielenden Routine mit Hilfe eines Strings übergeben (allerdings etwas anders, als dies in GW-Basic der Fall ist).
Die einzelnen Noten sind dabei, wie üblich, durch die Buchstaben

```
C, D, E, F, G, A, H
```

bezeichnet. Verwenden Sie Kleinbuchstaben, dann erklingen die Töne um eine Oktave höher. Hängt man ein » + « oder » – « an die Note, so verschiebt sich die Tonhöhe um einen Halbton nach oben beziehungsweise unten. Die chromatische Tonleiter sieht dann so aus:

```
C,C+,D,D+,E,F,F+,G,G+,A,A+,H,c
```

Zwischen jeder Note muß (im Gegensatz zu Basic) ein Leerzeichen oder ein Trennzeichen stehen. Außer den Buchstaben

```
L, O, P, S, T
```

(groß- oder kleingeschrieben) und den Zeichen » + «, » – « sowie den Ziffern »0« bis »9« und den sieben oben angeführten Notenbezeichnungen ist jedes Zeichen als Notentrennzeichen erlaubt; der Einfachheit halber wird man jedoch meist ein Leerzeichen setzen. Aufgrund dieser Festlegung läßt sich ein Taktbeginn innerhalb des Noten-Strings beispielsweise mit einem senkrechten Strich [Alt-179] verdeutlichen.
Eine besondere Bedeutung haben die beiden Zeichen »>« und »<«: Ab dem Größer-Zeichen werden alle Töne um eine Oktave höher und ab dem Kleiner-Zeichen eine Oktave tiefer ausgegeben. Darüber hinaus dürfen Sie die Noten-Strings formatfrei schreiben; das heißt, daß mehrere Trennzeichen hintereinander stets wie ein einziges wirken. Die Zeichenfolgen

```
'C E G c'
'C, E, G, c'
'C,E,G>C'
```

liefern daher ein völlig identisches Resultat (einen »Tusch«).
Die Zeichen »L«, »O«, »P«, »S«, »T«, » + « und » – « wirken als Steuerzeichen. Mit ihrer Hilfe bestimmen Sie die Taktart, das Tempo, die Grundoktave und auch das Transponieren in andere Tonarten. Ohne Angabe von Steuerzeichen verwendet die Unit »Music« feste Standardeinstellungen:

- Taktart: 4/4
- Taktlänge: 1000 Millisekunden
- Notenart: Viertelnoten
- Grundoktave: vierte Oktave; beginnend mit dem »tiefen C« (262 Hz)
- Spielweise: Legato (gebundenes Spiel)

Bevor die Steuerzeichen einer genaueren Betrachtung unterzogen werden, ist noch zu klären, warum die einzelnen Noten überhaupt getrennt zu schreiben sind. Die Ursache liegt darin, daß die Unit auch in der Lage ist, zweistimmige Pseudo-Akkorde zu spielen. Das Programm erreicht dies durch einen schnellen Wechsel zwischen zwei Tönen, wodurch ein tremoloartiger Doppelklang entsteht. Die Ausführung solcher Akkorde bewirken Sie durch das Nebeneinanderstellen zweier Notenzeichen. Der String

```
'C CE EG Ec'
```

läßt den Tusch zweistimmig ertönen. Beachten Sie bitte, daß Sie hierbei das »hohe C« mit einem Kleinbuchstaben schreiben müssen. Der String »E>C« spielt keinen Akkord, weil »>« nicht nur die Oktave erhöht, sondern auch als Trennzeichen fungiert.

Mit wenigen Steuerzeichen gibt der Programmierer (nicht nur) den Takt an

Mit dem Steuerzeichen »T« bestimmen Sie, wie viele einfache Noten in einem Takt zu spielen sind. Die Standardeinstellung »T4« setzt Viertelnoten, während »T8« beispielsweise Achtelnoten definiert. Das Steuerzeichen »L« legt die Taktlänge (Taktdauer) fest. So wird zum Beispiel mit »L500« eine Taktlänge von 500 Millisekunden eingestellt. »L0« oder auch nur »L« wählt die Standardtaktlänge von 1000 Millisekunden. Während dieser Zeit spielt die Routine so viele einfache Noten, wie mit dem Steuerzeichen »T« angegeben werden.
Wollen Sie anstatt einfacher einmal »gedehnte« Noten schreiben, so müssen Sie an das Notenzeichen (oder gegebenenfalls an das anschließende Zeichen » + « oder » – «) noch eine Ziffer anhängen. Die jeweilige Spieldauer einer gedehnten Note wird dann mit dieser Zahl multipliziert. Wenn also einfache Noten als Viertelnoten definiert sind (Standardeinstellung), dann werden durch Anhängen der Zahl 2 halbe Noten und durch Anhängen der Zahl 4 ganze Noten gespielt. Das Steuerzeichen »P« setzt Pausen. Diese lassen sich durch Hinzufügen von Ziffern – wie bei den gedehnten Noten – entsprechend verlängern.

Ob Legato oder Stakkato – die Music-Unit sorgt für den guten Ton

In der Standardeinstellung werden die Noten – falls keine Pausen dazwischenliegen – unmittelbar nacheinander (Legato) gespielt. Wenn Sie dagegen kürzere Töne erzeugen wollen, so bewerkstelligen Sie dies mit dem Steuerzeichen »S« (Stakkato) und einer darauffolgenden Zahl (1 bis 7). Mit »S1« erhalten die zu spielenden Töne sieben Achtel ihrer definierten Länge zugewiesen, mit »S2« sechs Achtel und so weiter.
Mit dem Steuerzeichen »O« setzen Sie die Oktavlage für die nachfolgend zu spielenden Noten. Erlaubt sind »O1« (niedrigste Oktave) bis »O7« (höchste Oktave). Zusätzlich läßt sich die Oktave mit dem Größer- und Kleiner-Zeichen ändern sowie durch Wechsel der Notenschreibweise von Groß- auf Kleinbuchstaben (und umgekehrt). Steuerzeichen ohne angehängte Zahlen setzen die nachfolgend gespielten Noten auf die entprechenden Standardeinstellungen zurück.
Eine weitere – sehr wirkungsvolle – Möglichkeit, Tonlagen zu verändern, ist das Transponieren: Erscheint ein » + « oder » – « nach einem Trennzeichen, so wird eine unmittelbar folgende Zahl als Transponierungsstufe interpretiert; das heißt, die Tonlage wird um eine der Zahl entsprechende Anzahl Halbtöne erhöht oder erniedrigt. Haben Sie daher einen Melodie-String in C-Dur vorbereitet und wollen Sie ihn später in F-Dur spielen, dann genügt es, wenn Sie vor der ersten Note » + 5« in den String einfügen, weil die Note F fünf Halbtöne über dem C liegt.
Listing 6 (»musik.pas«) zeigt ein Demoprogramm, das einige bekannte Melodien wiedergibt. Sofern Sie längere Musikstücke mit einem Programm wiedergeben möchten, gestal-

tet sich die Handhabung wesentlich flexibler, wenn Sie die Noten-Strings in einer eigenen Textdatei zur Verfügung stellen. Listing 7 (»spiele.pas«) stellt ein sehr einfaches Musikspielprogramm vor. Ein Beispiel für eine zugehörige Notendatei repräsentiert das Listing 8 (»bach2«). Nach dem Aufruf

```
spiele bach2
```

hören Sie das Präludium Nr. 2 aus dem Wohltemperierten Klavier – Johann Sebastian Bach möge uns verzeihen.

Turbo-schnelle Bildschirmausgabe ohne Nachbrenner und Rückenwind

Immer dann, wenn Sie die Turbo-Pascal-Unit »Crt« (mit »DirectVideo := true«) verwenden, werden alle über die Prozeduren »WriteLn« und »Write« abgewickelten Ausgaben unmittelbar in den Bildschirmspeicher übertragen. Als hinsichtlich des Laufzeitverhaltens verwöhnter Pascal-Programmierer könnte man daher der Meinung sein, daß diese Ausgaben mit größtmöglicher Geschwindigkeit vor sich gehen. Daß dem aber nicht so ist, zeigt die Programmroutine »PutStr«, mit der Strings mehr als dreimal so schnell auf den Bildschirm gezaubert werden wie mit dem Write-Kommando. Um den Bildschirm unter Verwendung der Write-Anweisung mit 24 Zeilen zu je 80 Zeichen zu füllen, benötigt ein mit 10 MHz getakteter AT 37 Millisekunden. Die Putstr-Prozedur dagegen bewältigt den Vorgang in nur 11 Millisekunden.

```
PutStr(x,y,Str);
```

wirkt dabei wie die Anweisungsfolge

```
GotoXY(x,y);
Write(Str);
```

die in Programmen sehr häufig auftritt. Die vom System verwaltete Cursor-Position wird mit »Putstr« jedoch nicht verändert. Daher läßt sich mit der Prozedur der gesamte Bildschirmbereich vollständig nutzen, weil beim Beschreiben der untersten rechten Bildschirmecke (Zeile 25, Spalte 80) kein unliebsamer Zeilenvorschub folgt.

Die Prozedur »PutStr« ist als Assemblermodul »__puts.asm« (Listing 9) realisiert. Die Routine verwendet die Variable »TextAttr« der Unit »Crt«. Damit ist sichergestellt, daß die Vorder- und Hintergrundfarben unverändert wiedergegeben werden. Durch die Definition der Prozedur als »proc far« sorgen Sie dafür, daß sich das Modul in eine Unit einbinden läßt. Im Interface-Teil einer solchen Unit müssen Sie lediglich die Variable »ScreenAdr« initialisieren. Der Wertparameter »Str« der Prozedur »PutStr« wird (vom Compiler unbemerkt) wie ein Var-Parameter behandelt – das zeitraubende Anlegen einer Kopie des übergebenen Parameters entfällt somit.

Falls Sie keinen Assembler besitzen, können Sie das erforderliche Objektmodul »__puts.obj« auch mit Hilfe des in Listing 10 gezeigten Hexdumps und dem Programm »SCheck« (DOS 1'90) abtippen. All diejenigen, für die der leichtfertige Umgang mit Speicherplatzressourcen ein schwerwiegendes Vergehen darstellt, dürfen beruhigt sein: Von den insgesamt 249 Byte wird nur ein Codeanteil von 58 Byte in die Turbo-Pascal-Unit eingebunden. Der Großteil des Objektmoduls besteht aus notwendigen Informationen für den Linker.

Listing 11 zeigt die Unit »gettext.pas« mit Routinen, die zum Lesen von Textdateien häufig benötigt werden. Eine praktische Anwendung dieser Unit zeigt das Programm »zeige.pas« (Listing 12). Es gestattet Ihnen ein bequemes Inspizieren von Textdateien mit Vorwärts- und Rückwärts-Scrollen und seitenorientiertem »Durchblättern«. Beim Programmstart übergeben Sie dazu einfach den Namen (einschließlich Dateierweiterung) der zu lesenden Datei:

```
zeige gettext.pas
```

Gerade dieser Anwendungsfall erfordert eine sehr schnelle Bildschirmausgabe, falls Sie mit einer erhöhten Anschlagswiederholrate Ihrer Tastatur arbeiten. AT-Besitzer, die noch mit der (etwas lahmen) Standardwiederholrate arbeiten, können Ihrer Tastatur mit dem folgenden kleinen Programm auf die Sprünge helfen (siehe auch DOS 11'89, Seite 240):

```
program speedup;
uses dos;
var r: registers;
begin
r.ax := $0305;
r.bx := $0100;
intr($16, r)
end.
```

Wenn Sie einmal versuchsweise die Putstr-Anweisungen durch Write-Anweisungen ersetzen (vor der For-Schleife wäre dazu die Anweisung »GotoXY(1,1)« einzufügen), kommt die Stunde der Wahrheit: Es zeigt sich die deutlich geringere Leistungsfähigkeit der Write-Prozedur beim schnellen Scrollen einer längeren Textdatei. Der Bildaufbau erfolgt ruckweise, und bei anhaltendem Drücken der Tasten [Cursor aufwärts] oder [Cursor abwärts] dürfte schon bald ein Piepsen als Kennzeichen für den vollen Tastaturpuffer an Ihren Nerven nagen.

Mit PutStr in der ursprünglichen Version geht dagegen der Bildaufbau »auf einen Schlag« vor sich, und beim andauernden Scrollen rutscht der Inhalt des Textbildschirms »wie geölt« nach oben oder unten. Wer daher die Absicht hegt, beispielsweise einen »Full-Screen-Editor« im Eigenbau zu entwickeln, wird mit dieser Assemblerroutine beim Programmieren der Bildschirmausgabe bestens bedient.

(Dr. Hermann Rechberger/zi)

Name	calendar.pas
Sprache	Turbo Pascal ab Version 4.Ø
Grafikkarte	jede
Besonderheiten	keine

```
[595]    1: { Programmname: CALENDAR.PAS }
[V94]    2: { (C)  Everts & Hagedorn     }
[ZB4]    3: {      DOS International     }
[C75]    4: { Autor: Hermann Rechberger  }
[ØØØ]    5:
[FX4]    6: (************ CALENDAR.PAS *************)
[ZB2]    7: UNIT CALENDAR;
[ØT3]    8: {$r-,l-,s-,i-,d-,f-,b-,v-}
[ØØØ]    9:
[SK1]   1Ø: INTERFACE
[ØØØ]   11:
```

```
[AØ4]   12: TYPE DateTyp = LONGINT;
[ØØØ]   13:
[6XØ]   14: CONST
[KM6]   15:     WeekdayName: ARRAY[Ø..7] OF STRING[1Ø]
[633]   16:                = ('ERROR', 'Montag',
[AS3]   17:                   'Dienstag', 'Mittwoch',
[ST3]   18:                   'Donnerstag','Freitag',
[C73]   19:                   'Samstag','Sonntag');
[ØØØ]   2Ø:
[WX5]   21:       MonthName: ARRAY[1..12] OF STRING[9]
[FD3]   22:                = ('Januar', 'Februar',
[UL3]   23:                   'März', 'April', 'Mai',
[6V3]   24:                   'Juni', 'Juli', 'August',
[7P3]   25:                   'September', 'Oktober',
[FN3]   26:                   'November', 'Dezember');
[ØØØ]   27:
[X66]   28: FUNCTION Leapyear(y: WORD): BOOLEAN;
[CØ5]   29: FUNCTION PackDate(year: WORD;
[WE4]   3Ø:                   month, day: BYTE): DateTyp;
[KB4]   31: FUNCTION today: DateTyp;
[2T6]   32: FUNCTION DateError(dt: DateTyp): BOOLEAN;
```

```
[ZB5]    33: FUNCTION Day(dt: DateTyp): BYTE;
[1N5]    34: FUNCTION Month(dt: DateTyp): BYTE;
[SH5]    35: FUNCTION Year(dt: DateTyp): WORD;
[NA6]    36: FUNCTION DayofWeek(dt: DateTyp): BYTE;
[4Ø5]    37: FUNCTION AddDate(dt: DateTyp;
[DC3]    38:                  n: longint): DateTyp;
[YN5]    39: FUNCTION DiffDate(d1, d2: DateTyp):
[BG1]    4Ø:                                longint;
[HF6]    41: FUNCTION MakeDate(s: STRING): DateTyp;
[S86]    42: FUNCTION Easter(year: WORD): DateTyp;
[ØØØ]    43:
[ØØØ]    44:
[P84]    45: IMPLEMENTATION USES dos;
[ØØØ]    46:
[CA2]    47: CONST ya = 17ØØ;
[M91]    48:       ye = 2199;
[ØØØ]    49:
[4C5]    5Ø: TYPE hcon = ARRAY[1..12] OF BYTE;
[ØØØ]    51:
[854]    52:      htyp = RECORD CASE BYTE OF
[2Z2]    53:               Ø: (date: DateTyp);
[DA3]    54:               1: (d, m: BYTE; y: WORD)
[ZNØ]    55:             END;
[ØØØ]    56:
[ØØØ]    57:
[X66]    58: FUNCTION Leapyear(y: WORD): BOOLEAN;
[MUØ]    59: BEGIN
[1X2]    6Ø:   Leapyear := FALSE;
[4W3]    61:   IF (y AND 3 <> Ø) THEN exit
[8C5]    62:   ELSE Leapyear := (y MOD 1ØØ <> Ø) OR
[7V1]    63:                    (y MOD 4ØØ = Ø)
[ZNØ]    64: END;
[ØØØ]    65:
[CØ5]    66: FUNCTION PackDate(year: WORD;
[WE4]    67:                   month, day: BYTE): DateTyp;
[BØ2]    68: VAR h: htyp;
[MUØ]    69: BEGIN
[9F2]    7Ø:   WITH h DO BEGIN
[Ø24]    71:     IF year > 1ØØ THEN y := year
[B33]    72:                   ELSE y := 19ØØ + year;
[WT1]    73:     m := month;
[C71]    74:     d := day;
[WL2]    75:     PackDate := date
[4MØ]    76:   END
[ZNØ]    77: END;
[ØØØ]    78:
[ZB5]    79: FUNCTION Day(dt: DateTyp): BYTE;
[BØ2]    8Ø: VAR h: htyp;
[MUØ]    81: BEGIN
[9F2]    82:   WITH h DO BEGIN
[KU1]    83:     date := dt;
[Y51]    84:     Day := d
[4MØ]    85:   END
[ZNØ]    86: END;
[ØØØ]    87:
[1N5]    88: FUNCTION Month(dt: DateTyp): BYTE;
[BØ2]    89: VAR h: htyp;
[MUØ]    9Ø: BEGIN
[9F2]    91:   WITH h DO BEGIN
[KU1]    92:     date := dt;
[FK1]    93:     Month := m
[4MØ]    94:   END
[ZNØ]    95: END;
[ØØØ]    96:
[SH5]    97: FUNCTION Year(dt: DateTyp): WORD;
[BØ2]    98: VAR h: htyp;
[MUØ]    99: BEGIN
[9F2]   1ØØ:   WITH h DO BEGIN
[KU1]   1Ø1:     date := dt;
[BC1]   1Ø2:     Year := y
[4MØ]   1Ø3:   END
[ZNØ]   1Ø4: END;
[ØØØ]   1Ø5:
[KB4]   1Ø6: FUNCTION today: DateTyp;
[BØ2]   1Ø7: VAR h: htyp;
[ZK3]   1Ø8:     month, day, dow: WORD;
[MUØ]   1Ø9: BEGIN
[9F2]   11Ø:   WITH h DO BEGIN
[LA4]   111:     getdate(y, month, day, dow);
[WT1]   112:     m := month;
[C71]   113:     d := day;
[BØ2]   114:     today := date
[4MØ]   115:   END
[ZNØ]   116: END;
[ØØØ]   117:
[BW4]   118: FUNCTION DayspMonth(y: WORD;
[DC2]   119:                     m: BYTE): BYTE;
[P13]   12Ø: (* days per month *)
[U65]   121: CONST dpm: hcon = (31, 28, 31, 3Ø, 31, 3Ø,
[ZN2]   122:                    31, 31, 3Ø, 31, 3Ø, 31);
[MUØ]   123: BEGIN
[TH3]   124:   DayspMonth := dpm[m];
[DX2]   125:   IF m <> 2 THEN exit
[EM6]   126:   ELSE IF Leapyear(y) THEN DayspMonth := 29
[ZNØ]   127: END;
[ØØØ]   128:
```

```
[2T6]   129: FUNCTION DateError(dt: DateTyp): BOOLEAN;
[TN2]   13Ø: VAR hdate: htyp;
[MUØ]   131: BEGIN
[M13]   132:   DateError := TRUE;
[H53]   133:   WITH hdate DO BEGIN
[KU1]   134:     date := dt;
[XH2]   135:     IF (m < 1) OR (m > 12)
[LK4]   136:       OR (y < ya) OR (y > ye) OR (d < 1)
[VH5]   137:       OR (d > DayspMonth(y, m)) THEN exit
[WV3]   138:     ELSE DateError := FALSE
[4MØ]   139:   END
[ZNØ]   14Ø: END;
[ØØØ]   141:
[NA6]   142: FUNCTION DayofWeek(dt: DateTyp): BYTE;
[TH3]   143: (* Wochentag bestimmen
[KY5]   144: bei ungÜltigem Datum: DayofWeek = Ø,
[WT4]   145: sonst 1..7 (Montag..Sonntag) *)
[8R2]   146: CONST kenn: hcon
[784]   147:   = (8, 9, 2, 3, 3, 4, 4, 5, 6, 6, 7, 7);
[TN2]   148: VAR hdate: htyp;
[AF1]   149:     y1: WORD;
[WN1]   15Ø:     n, m1: BYTE;
[MUØ]   151: BEGIN
[ST5]   152:   IF DateError(dt) THEN DayofWeek := Ø
[AY3]   153:   ELSE WITH hdate DO BEGIN
[KU1]   154:     date := dt;
[G11]   155:     m1 := m;
[S31]   156:     y1 := y;
[WW2]   157:     IF m <= 2 THEN BEGIN
[RH1]   158:       m1 := m + 12;
[VE1]   159:       y1 := y - 1
[ZNØ]   16Ø:     END;
[M94]   161:     n := (d + 2 * m1 + kenn[m] + y1 +
[YM3]   162:          (y1 DIV 4) - (y1 DIV 1ØØ) +
[253]   163:          (y1 DIV 4ØØ) + 2) MOD 7;
[FF4]   164:     IF n > 1 THEN DayofWeek := n - 1
[YK3]   165:              ELSE DayofWeek := n + 6
[4MØ]   166:   END
[ZNØ]   167: END;
[ØØØ]   168:
[6V5]   169: PROCEDURE incdate(VAR dt: DateTyp);
[TN2]   17Ø: VAR hdate: htyp;
[MUØ]   171: BEGIN
[H53]   172:   WITH hdate DO BEGIN
[KU1]   173:     date := dt;
[RE5]   174:     IF d < DayspMonth(y, m) THEN inc(d)
[WP1]   175:     ELSE BEGIN
[CYØ]   176:       d := 1;
[LS2]   177:       IF m < 12 THEN inc(m)
[WP1]   178:       ELSE BEGIN
[YYØ]   179:         m := 1;
[XYØ]   18Ø:         inc(y)
[4MØ]   181:       END
[ZNØ]   182:     END;
[ØF1]   183:     dt := date
[4MØ]   184:   END
[ZNØ]   185: END;
[ØØØ]   186:
[9U5]   187: PROCEDURE decdate(VAR dt: DateTyp);
[TN2]   188: VAR hdate: htyp;
[MUØ]   189: BEGIN
[H53]   19Ø:   WITH hdate DO BEGIN
[KU1]   191:     date := dt;
[WR2]   192:     IF d > 1 THEN dec(d)
[WP1]   193:     ELSE BEGIN
[LM4]   194:       IF m > 1 THEN dec(m) ELSE BEGIN
[NZØ]   195:         m := 12;
[SXØ]   196:         dec(y)
[ZNØ]   197:       END;
[463]   198:       d := DayspMonth(y, m)
[ZNØ]   199:     END;
[ØF1]   2ØØ:     dt := date
[4MØ]   2Ø1:   END
[ZNØ]   2Ø2: END;
[ØØØ]   2Ø3:
[4Ø5]   2Ø4: FUNCTION AddDate(dt: DateTyp;
[DC3]   2Ø5:                  n: longint): DateTyp;
[MK2]   2Ø6: VAR i: longint;
[U72]   2Ø7:     hdt: DateTyp;
[MUØ]   2Ø8: BEGIN
[MG1]   2Ø9:   hdt := dt;
[RF4]   21Ø:   IF n > Ø THEN FOR i := 1 TO n DO
[V22]   211:     incdate(hdt)
[9N5]   212:   ELSE FOR i := 1 TO n DO decdate(hdt);
[B62]   213:   AddDate := hdt
[ZNØ]   214: END;
[ØØØ]   215:
[YN5]   216: FUNCTION DiffDate(d1, d2: DateTyp):
[BG1]   217:                                longint;
[LG3]   218: VAR hd1, hd2: DateTyp;
[7W1]   219:     n: longint;
[4S2]   22Ø:     sign: shortint;
[MUØ]   221: BEGIN
[G63]   222:   IF d2 >= d1 THEN BEGIN
[2P1]   223:     sign := +1;
[2A1]   224:     hd1 := d1;
```

Listing 1. Die Unit »calendar.pas« bietet Ihnen Funktionen für Datumsberechnungen

```
[R41]  225:      hd2 := d2
[ØC2]  226:    END ELSE BEGIN
[6P1]  227:      sign := -1;
[4A1]  228:      hd1 := d2;
[N41]  229:      hd2 := d1
[ZNØ]  23Ø:    END;
[XYØ]  231:    n := Ø;
[AU3]  232:    WHILE hd1 <> hd2 DO BEGIN
[L51]  233:      inc(n);
[FY1]  234:      incdate(hd1)
[ZNØ]  235:    END;
[UØ3]  236:    DiffDate := sign * n
[ZNØ]  237: END;
[ØØØ]  238:
[HF6]  239: FUNCTION MakeDate(s: STRING): DateTyp;
[Z41]  24Ø: LABEL 1;
[RY2]  241: CONST endchr = #255;
[PT3]  242: VAR i, j, len, p, y: WORD;
[GL3]  243:     n: ARRAY[1..3] OF WORD;
[YE1]  244:     hØ: htyp;
[UL1]  245:     m, d: BYTE;
[D61]  246:     c: CHAR;
[SC3]  247: FUNCTION next: CHAR;
[773]  248: {i, len, s : global}
[MUØ]  249: BEGIN
[951]  25Ø:   inc(i);
[1K4]  251:   IF i <= len THEN next := s[i]
[L73]  252:                 ELSE next := endchr
[ZNØ]  253: END;
[Ø14]  254: FUNCTION look: BOOLEAN;
[773]  255: {i, len, s : global}
[MUØ]  256: BEGIN
[C62]  257:   IF i < len THEN
[5C4]  258:     look := s[i + 1] IN ['Ø'..'9']
[ØX2]  259:               ELSE look := FALSE
[ZNØ]  26Ø: END;
[ZX3]  261: FUNCTION number: WORD;
[S92]  262: {p: global}
[SC2]  263: VAR j, n: WORD;
[MUØ]  264: BEGIN
[XYØ]  265:   n := Ø;
[ML2]  266:   FOR j := 1 TO p DO
[9Z1]  267:     IF look THEN
[HS3]  268:       n := n * 1Ø + ORD(next) - 48;
[XU1]  269:   number := n
[ZNØ]  27Ø: END;
[XU2]  271: BEGIN (* MakeDate *)
[DM2]  272:   len := length(s);
[LYØ]  273:   i := Ø;
[LC3]  274:   FOR j := 1 TO 3 DO BEGIN
[XS3]  275:     WHILE NOT look DO BEGIN
[UF1]  276:       c := next;
[EP3]  277:       IF c = endchr THEN GOTO 1
[ZNØ]  278:     END;
[LW4]  279: 1:  IF j = 3 THEN p := 4 ELSE p := 2;
[HN2]  28Ø:     n[j] := number;
[ZNØ]  281:   END;
[AG2]  282:   hØ.date := today;
[ZH1]  283:   d := n[1];
[KK1]  284:   m := n[2];
[BL1]  285:   y := n[3];
[UY4]  286:   IF (y = Ø) AND (m = Ø) AND (d = Ø) THEN
[9Ø6]  287:     MakeDate := hØ.date {leere Eingabe}
[WP1]  288:   ELSE BEGIN
[2Z4]  289:     IF (y = Ø) AND (m > Ø) AND (d > Ø) THEN
[F71]  29Ø:       y := hØ.y
[YC4]  291:     ELSE IF (y = Ø) AND (m = Ø) AND
[DH2]  292:       (d > Ø) THEN BEGIN
[ØD1]  293:       y := hØ.y;
[Z51]  294:       m := hØ.m
[ZNØ]  295:     END;
[9D4]  296:     MakeDate := PackDate(y, m, d)
[4MØ]  297:   END
[ZNØ]  298: END;
[ØØØ]  299:
[S86]  3ØØ: FUNCTION Easter(year: WORD): DateTyp;
[RY5]  3Ø1: {Datum des Ostersonntags von "year"
[ZU4]  3Ø2:  (17ØØ bis 2199) bestimmen.        }
[W34]  3Ø3: VAR a, b, c, d, e, m, n: WORD;
[US2]  3Ø4:     month, day: BYTE;
[MUØ]  3Ø5: BEGIN
[HV4]  3Ø6:   IF (year < ya) OR (year > ye) THEN
[MUØ]  3Ø7:   BEGIN
[4R1]  3Ø8:     month := Ø;
[Ø91]  3Ø9:     day := Ø;
[ASØ]  31Ø:     exit
[ZNØ]  311:   END;
[TØ3]  312:   CASE year DIV 1ØØ OF
[WB4]  313:     17:      BEGIN m := 23; n := 3 END;
[1C4]  314:     18:      BEGIN m := 23; n := 4 END;
[CG4]  315:     19, 2Ø: BEGIN m := 24; n := 5 END;
[364]  316:     21:      BEGIN m := 24; n := 6 END
[ZNØ]  317:   END;
[9C2]  318:   a := year MOD 19;
[392]  319:   b := year MOD 4;
[H92]  32Ø:   c := year MOD 7;
[YU2]  321:   d := (19 * a + m) MOD 3Ø;
[KT4]  322:   e := (2 * b + 4 * c + 6 * d + n) MOD 7;
[CF2]  323:   day := (22 + d + e);
[DR1]  324:   month := 3;
[F73]  325:   IF day > 31 THEN BEGIN
[DF2]  326:     day := (d + e - 9);
[GR1]  327:     month := 4;
[PY2]  328:     IF (d = 28) AND (e = 6)
[MD3]  329:                 AND (a > 1Ø) THEN BEGIN
[9Y3]  33Ø:       IF (day = 26) THEN day := 19;
[HU3]  331:       IF (day = 25) THEN day := 18
[4MØ]  332:     END
[ZNØ]  333:   END;
[4U5]  334:   Easter := PackDate(year, month, day)
[ZNØ]  335: END;
[ØØØ]  336:
[KNØ]  337: END.
```

Listing 1. Die Unit »calendar.pas« bietet Ihnen Funktionen für Datumsberechnungen

Name	gebdat.pas
Sprache	Turbo Pascal ab Version 4.Ø
Grafikkarte	jede
Besonderheiten	keine

```
[G25]    1: { Programmname: GEBDAT.PAS   }
[V94]    2: { (C)  Everts & Hagedorn     }
[ZB4]    3: {      DOS International     }
[C75]    4: { Autor: Hermann Rechberger  }
[ØØØ]    5:
[7R2]    6: PROGRAM GEBDAT;
[ØT3]    7: {$R-,L-,S-,I-,D-,F-,B-,V-}
[V73]    8: USES crt, calendar;
[ØØØ]    9:
[RNØ]   1Ø: VAR
[Y23]   11:   GebDatum: DateTyp;
[RF3]   12:   Eingabe: STRING[24];
[Ø22]   13:   ok: BOOLEAN;
[ØØØ]   14:
[MUØ]   15: BEGIN
[DB1]   16:   ClrScr;
[P31]   17:   REPEAT
[P31]   18:     REPEAT
[1U5]   19:       Write('Geben Sie Ihr Geburtsdatum '+
[Y41]   2Ø:             'ein: ');
[DH2]   21:       ReadLn(Eingabe);
[6T3]   22:       IF Eingabe = '' THEN halt;
[AX4]   23:       GebDatum := MakeDate(Eingabe);
[WW4]   24:       ok := NOT DateError(GebDatum);
[6K3]   25:       IF NOT ok THEN Write(#7)
[2H1]   26:     UNTIL ok;
[XF1]   27:     WriteLn;
[2F5]   28:     IF GebDatum >= today THEN WriteLn(
[PZ5]   29:       'Bei diesem Geburtsdatum muß ein '+
[6D3]   3Ø:       'Irrtum vorliegen !')
[U82]   31:     ELSE WriteLn(
[643]   32:       'Sie sind an einem ',
[UM5]   33:       WeekdayName[DayofWeek(GebDatum)],
[VA2]   34:       ' geboren und ',
[V44]   35:       DiffDate(GebDatum, today),
[LY1]   36:       ' Tage alt.');
[2E1]   37:     WriteLn
[312]   38:   UNTIL FALSE
[KNØ]   39: END.
```

Listing 2. Das Programm »gebdat.pas« nennt Ihnen Ihr Alter anders als gewohnt

Name	kalender.pas
Sprache	Turbo Pascal ab Version 4.Ø
Grafikkarte	jede
Besonderheiten	keine

```
[N95]    1: { Programmname: KALENDAR.PAS }
[V94]    2: { (C)  Everts & Hagedorn     }
[ZB4]    3: {      DOS International     }
[C75]    4: { Autor: Hermann Rechberger  }
[ØØØ]    5:
```

```
[6Z2]    6: PROGRAM KALENDER;
[ØT3]    7: {$R-,L-,S-,I-,D-,F-,B-,V-}
[V73]    8: USES Crt, Calendar;
[ØØØ]    9:
[6XØ]   1Ø: CONST
[5E6]   11:   ZahlFF = 5; {Zahl der festen Feiertage }
[ØG6]   12:   ZahlBF = 5; {Zahl der beweglichen Feste}
[ØØØ]   13:
[396]   14: TYPE FT = RECORD Monat, Tag: BYTE END;
[ØØØ]   15:
[6XØ]   16: CONST
[ZS6]   17:   FesteFeiertage: ARRAY[1..ZahlFF] OF FT =
[UX5]   18:   ((Monat:Ø1; Tag: Ø1), { Neujahr          }
[A35]   19:    (Monat:Ø5; Tag: Ø1), { 1. Mai           }
[U86]   2Ø:    (Monat:12; Tag: 25), { 1. Weihnachtstag}
[Y86]   21:    (Monat:12; Tag: 26), { 2. Weihnachtstag}
[SS5]   22:    (Monat:12; Tag: 31));{ Silvester       }
[ØØØ]   23:
[WZ6]   24:   BewFest: ARRAY[1..ZahlBF] OF STRING[16]
[L33]   25:     = ('Ostersonntag: ',
[XK2]   26:        'Ostermontag: ',
[5H2]   27:        'Himmelf.-Tag: ',
[F73]   28:        'Pfingstsonntag: ',
[R63]   29:        'Pfingstmontag: ');
[ØØØ]   3Ø:
[RU6]   31:   Osterabstand: ARRAY[1..ZahlBF] OF LONGINT
[852]   32:       = (Ø, 1, 38, 1Ø, 1);
[ØØØ]   33:
[ZH2]   34: VAR i: INTEGER;
[4X1]   35:     Jahr: WORD;
[GL2]   36:     Datum: DateTyp;
[542]   37:     Mt, Tg: BYTE;
[ØØØ]   38:
[E85]   39: PROCEDURE PutDate(dt: DateTyp);
[MUØ]   4Ø: BEGIN
[LH3]   41:   WriteLn(Day(dt):2, '.',
[LN3]   42:     MonthName[Month(dt)])
[ZNØ]   43: END;
[ØØØ]   44:
[MUØ]   45: BEGIN
[DB1]   46:   ClrScr;
[UF3]   47:   IF paramcount = Ø THEN
[Z43]   48:     Jahr := Year(today)
[WP1]   49:   ELSE BEGIN
[TZ3]   5Ø:     val(paramstr(1), Jahr, i);
[UK5]   51:     IF Jahr < 1ØØ THEN Jahr := Jahr + 19ØØ;
[YA5]   52:     IF (Jahr < 17ØØ) OR (Jahr > 2199) THEN
[MUØ]   53:     BEGIN
[MX1]   54:       WriteLn(#7);
[RPØ]   55:       halt
[4MØ]   56:     END
[ZNØ]   57:   END;
[YW6]   58:   WriteLn('Kalenderübersicht für ', Jahr);
[XF1]   59:   WriteLn;
[XF1]   6Ø:   WriteLn;
[DG4]   61:   WriteLn('Feste Feiertage:');
[XF1]   62:   WriteLn;
[ØL3]   63:   FOR I := 1 TO ZahlFF DO
[AC5]   64:   WITH FesteFeiertage[I] DO BEGIN
[RH5]   65:     Datum := PackDate(Jahr, Monat, Tag);
[8C6]   66:     Write(WeekdayName[DayofWeek(Datum)]
[671]   67:       :1Ø, ' * ');
[YP2]   68:     PutDate(Datum);
[ZNØ]   69:   END;
[XF1]   7Ø:   WriteLn;
[XF1]   71:   WriteLn;
[1U4]   72:   WriteLn('Bewegliche Feste:');
[XF1]   73:   WriteLn;
[8H3]   74:   Datum := Easter(Jahr);
[4E4]   75:   FOR I := 1 TO ZahlBF DO BEGIN
[VR3]   76:     Datum := AddDate(Datum,
[T33]   77:                      Osterabstand[I]);
[AG3]   78:     Write(BewFest[I]:16);
[YP2]   79:     PutDate(Datum);
[ZNØ]   8Ø:   END;
[KNØ]   81: END.
```

Listing 3. Mit dem Programm »kalender.pas« erzeugen Sie sich einfache Kalenderübersichten

Name	abergl.pas
Sprache	Turbo Pascal ab Version 4.Ø
Grafikkarte	jede
Besonderheiten	keine

```
[925]    1: { Programmname: ABERGL.PAS    }
[V94]    2: { (C)  Everts & Hagedorn      }
[ZB4]    3: {      DOS International      }
[C75]    4: { Autor: Hermann Rechberger   }
[ØØØ]    5:
[CC3]    6: PROGRAM ABERGLAUBE;
[ØT3]    7: {$R-,L-,S-,I-,D-,F-,B-,V-}
[V73]    8: USES Crt, Calendar;
[ØØØ]    9:
[VL2]   1Ø: VAR Jahr: WORD;
[GL2]   11:     Datum: DateTyp;
[NØ2]   12:     Monat: BYTE;
[ØØØ]   13:
[MUØ]   14: BEGIN
[DB1]   15:   ClrScr;
[AF1]   16:   WriteLn(
[PU5]   17:     'Jahr | Monate mit Freitag den 13.');
[AF1]   18:   WriteLn(
[T8G]   19:     '-----+-----------------------------------');
[9Ø5]   2Ø:   FOR Jahr := 199Ø TO 2Ø1Ø DO BEGIN
[C93]   21:     Write(Jahr, ' | ');
[5B4]   22:     FOR Monat := 1 TO 12 DO BEGIN
[BD5]   23:       Datum := PackDate(Jahr, Monat, 13);
[1P4]   24:       IF (DayOfWeek(Datum) = 5) THEN
[NN4]   25:         Write(MonthName[Monat], '  ')
[ZNØ]   26:     END;
[2E1]   27:     WriteLn
[4MØ]   28:   END
[KNØ]   29: END.
```

Listing 4. Vor einem Freitag, den 13., warnt Sie das Programm »abergl.pas«

Name	music.pas
Sprache	Turbo Pascal ab Version 4.Ø
Grafikkarte	jede
Besonderheiten	keine

```
[B35]    1: { Programmname: MUSIC.PAS     }
[V94]    2: { (C)  Everts & Hagedorn      }
[ZB4]    3: {      DOS International      }
[C75]    4: { Autor: Hermann Rechberger   }
[ØØØ]    5:
[RU4]    6: (************** MUSIC.PAS **************)
[112]    7: UNIT music;
[ØT3]    8: {$r-,l-,s-,i-,d-,f-,b-,v-}
[ØØØ]    9:
[SK1]   1Ø: INTERFACE
[ØØØ]   11:
[KF6]   12: PROCEDURE playnote(freq, dauer: WORD);
[WB4]   13: PROCEDURE play(s: STRING);
[ØØØ]   14:
[194]   15: IMPLEMENTATION USES crt;
[ØØØ]   16:
[CE3]   17: CONST normtakt = 1ØØØ;
[RN4]   18:       taktdauer: WORD = normtakt;
[H23]   19:       stakkato: WORD = Ø;
[VP2]   2Ø:       oktave: WORD = 4;
[KP2]   21:       transp: WORD = Ø;
[7D2]   22:       takt:   WORD = 4;
[ØØØ]   23:
[KF6]   24: PROCEDURE playnote(freq, dauer: WORD);
[MUØ]   25: BEGIN
[VZ1]   26:   sound(freq);
[352]   27:   delay(dauer);
[8E1]   28:   nosound
[ZNØ]   29: END;
[ØØØ]   3Ø:
[WB4]   31: PROCEDURE play(s: STRING);
[ØØØ]   32:
[PN2]   33: CONST space = ' ';
[N23]   34:       ziffern = ['Ø'..'9'];
[WF4]   35:       noten   = ['C','D','E','F','G','A',
[NY3]   36:          'H','c','d','e','f','g','a','h'];
[AP5]   37:     steuerzeichen = ['P','L','T','S','O',
[ZF3]   38:              '+','-','p','l','t','s','o'];
[ØØØ]   39:
[7F5]   4Ø: VAR dauer, freq1, freq2, duration,
[GY3]   41:     pause, i, len, leng: WORD;
[XX2]   42:     c, s1, s2, s3, s4: CHAR;
[FT2]   43:     gruppe: STRING;
[N74]   44:     geteilt, fertig: BOOLEAN;
[XG3]   45:     trenner: SET OF CHAR;
[ØØØ]   46:
[ZA6]   47: FUNCTION freq(note, vorz: CHAR): WORD;
[253]   48: CONST tonzahl = 85;
[P36]   49: CONST fr: ARRAY[1..tonzahl] OF WORD =
[C53]   5Ø:         (            33,   35,   37,   39,
[LB3]   51:          41,   44,   46,   49,   52,   55,
[BD3]   52:          58,   62,   65,   69,   73,   78,
```

Listing 5. Die Unit »music.pas« läßt den Computer nach Ihren Noten spielen

```
[DE3]   53:          82,   87,   92,   98,  1Ø4,  11Ø,
[4P3]   54:         117,  123,  131,  139,  147,  156,
[2R3]   55:         165,  175,  185,  196,  2Ø8,  22Ø,
[WP3]   56:         233,  247,  262,  277,  294,  311,
[8P3]   57:         33Ø,  349,  37Ø,  392,  415,  44Ø,
[8T3]   58:         466 , 494,  523,  554,  587,  622,
[RT3]   59:         659 , 698,  74Ø,  784,  831,  88Ø,
[PW3]   6Ø:         932,  988, 1Ø47, 11Ø9, 1175, 1245,
[X44]   61:        1319, 1397, 148Ø, 1568, 1661, 176Ø,
[564]   62:        1865, 1976, 2Ø93, 2217, 2349, 2489,
[S44]   63:        2637, 2794, 296Ø, 3136, 3322, 352Ø,
[A52]   64:        3729, 3951, 4186);
[ØØØ]   65:
[5X2]   66: VAR  index: WORD;
[ØØØ]   67:
[MUØ]   68: BEGIN
[R55]   69:   index := (oktave - 1) * 12 + transp;
[ØØ2]   7Ø:   CASE note OF
[VC2]   71:     'C': inc(index);
[7U2]   72:     'D': inc(index,  3);
[GU2]   73:     'E': inc(index,  5);
[PU2]   74:     'F': inc(index,  6);
[ZU2]   75:     'G': inc(index,  8);
[RU2]   76:     'A': inc(index, 1Ø);
[KV2]   77:     'H': inc(index, 12);
[6V2]   78:     'c': inc(index, 13);
[FV2]   79:     'd': inc(index, 15);
[SV2]   8Ø:     'e': inc(index, 17);
[YV2]   81:     'f': inc(index, 18);
[BV2]   82:     'g': inc(index, 2Ø);
[ZU2]   83:     'a': inc(index, 22);
[UV2]   84:     'h': inc(index, 24);
[ZNØ]   85:   END;
[Ø22]   86:   CASE vorz OF
[LA2]   87:     '+': inc(index);
[B72]   88:     '-': dec(index)
[ZNØ]   89:   END;
[HX3]   9Ø:   IF index IN [1..tonzahl]
[NR3]   91:     THEN freq := fr[index]
[7V2]   92:     ELSE freq := 2ØØØØ;
[842]   93: END; (* freq *)
[ØØØ]   94:
[9D3]   95: FUNCTION zahl: WORD;
[YU2]   96: (* gruppe: global *)
[2D2]   97: VAR i, z: WORD;
[MUØ]   98: BEGIN
[NZØ]   99:   z := Ø;
[5Z2]  1ØØ:   FOR i := 1 TO leng DO
[MW4]  1Ø1:     IF gruppe[i] IN ziffern THEN
[ZV4]  1Ø2:       z := z * 1Ø + ord(gruppe[i]) - 48;
[TC1]  1Ø3:   zahl := z
[ZNØ]  1Ø4: END;
[ØØØ]  1Ø5:
[RX2]  1Ø6: PROCEDURE accord;
[8B5]  1Ø7: (* duration, freq1, freq2: global *)
[LL2]  1Ø8: CONST accd     = 4Ø;
[GØ2]  1Ø9:       zuschl   = 2Ø;
[PC2]  11Ø: VAR i, n: WORD;
[MUØ]  111: BEGIN
[P65]  112:   n := duration DIV (accd + zuschl);
[XF3]  113:   FOR i := 1 TO n DO BEGIN
[HK3]  114:     playnote(freq1, accd);
[3D3]  115:     playnote(freq2, accd)
[ZNØ]  116:   END;
[L22]  117:   delay(pause)
[ZNØ]  118: END;
[ØØØ]  119:
[282]  12Ø: BEGIN (* play *)
[PH4]  121:   trenner := [#Ø..#255] - noten -
[854]  122:              steuerzeichen - ziffern;
[ZZ2]  123:   len := length(s) + 1;
[BH2]  124:   fertig := false;
[ZF4]  125:   dauer := taktdauer DIV takt;
[4V1]  126:   gruppe := '';
[AU3]  127:   FOR i := 1 TO len DO BEGIN
[T14]  128:     IF keypressed THEN exit;
[4L1]  129:     c := s[i];
[626]  13Ø:     geteilt := (c IN trenner) OR (i = len);
[GP5]  131:     IF NOT geteilt OR fertig THEN BEGIN
[1L5]  132:       IF (geteilt AND fertig) THEN BEGIN
[BH2]  133:         fertig := false;
[TU3]  134:         leng := length(gruppe);
[KK2]  135:         s1 := gruppe[1];
[8W4]  136:         IF leng > 1 THEN s2 := gruppe[2]
[8K2]  137:                     ELSE s2 := space;
[MW4]  138:         IF leng > 2 THEN s3 := gruppe[3]
[CK2]  139:                     ELSE s3 := space;
[ØX4]  14Ø:         IF leng > 3 THEN s4 := gruppe[4]
[GK2]  141:                     ELSE s4 := space;
[1Z3]  142:         duration := zahl * dauer;
[L43]  143:         IF duration = Ø THEN
[S23]  144:           duration := dauer;
[2P5]  145:         pause := duration DIV 8 * stakkato;
[VU4]  146:         duration := duration - pause;
[2N3]  147:         IF (s1 IN noten) AND NOT
[K43]  148:            (s2 IN noten) AND NOT
[CP3]  149:            (s3 IN noten) THEN BEGIN
[3B5]  15Ø:         playnote(freq(s1, s2), duration);
[L22]  151:         delay(pause)
[4MØ]  152:       END
[HU3]  153:       ELSE IF (s1 IN noten) AND
[8P3]  154:               (s2 IN noten) THEN BEGIN
[PU3]  155:         freq1 := freq(s1, space);
[1A3]  156:         freq2 := freq(s2, s3);
[PØ1]  157:         accord
[4MØ]  158:       END
[R83]  159:       ELSE IF (s1 IN noten)
[8B4]  16Ø:          AND (s3 IN noten) THEN BEGIN
[X93]  161:         freq1 := freq(s1, s2);
[3A3]  162:         freq2 := freq(s3, s4);
[PØ1]  163:         accord
[4MØ]  164:       END
[WP1]  165:       ELSE BEGIN
[ØY2]  166:         CASE upcase(s1) OF
[DK4]  167:           'P': delay(duration + pause);
[KG1]  168:             (* pause *)
[5F1]  169:           'L': BEGIN
[VY2]  17Ø:                  taktdauer := zahl;
[S73]  171:                  IF taktdauer < 1 THEN
[XH3]  172:                    taktdauer := normtakt
[ZNØ]  173:                END;
[XF1]  174:           'T': BEGIN
[622]  175:                  takt := zahl;
[9D2]  176:                  IF takt < 1 THEN
[NA1]  177:                    takt := 4
[ZNØ]  178:                END;
[UF1]  179:           'S': BEGIN
[4U2]  18Ø:                  stakkato := zahl;
[D53]  181:                  IF stakkato > 7 THEN
[T22]  182:                    stakkato := 7
[ZNØ]  183:                END;
[L93]  184:             (* transpositionen *)
[343]  185:           '+': transp :=  zahl;
[P43]  186:           '-': transp := -zahl;
[EF1]  187:           'O': BEGIN
[8H2]  188:                  oktave := zahl;
[7R2]  189:                  IF oktave < 1 THEN
[FR1]  19Ø:                    oktave := 4
[4MØ]  191:                END
[V32]  192:         END; (* CASE *)
[ZF4]  193:         dauer := taktdauer DIV takt;
[Ø52]  194:       END; (* ELSE *)
[ER1]  195:       gruppe := ''
[4MØ]  196:     END
[WP1]  197:     ELSE BEGIN
[G55]  198:     (* IF NOT geteilt OR NOT fertig *)
[TF2]  199:       fertig := true;
[W13]  2ØØ:       gruppe := gruppe + c
[ZNØ]  2Ø1:     END;
[675]  2Ø2:   END; (* IF NOT geteilt OR fertig *)
[Y91]  2Ø3:   CASE c OF
[2G5]  2Ø4:     '>': IF oktave < 7 THEN inc(oktave);
[CC5]  2Ø5:     '<': IF oktave > 1 THEN dec(oktave)
[4MØ]  2Ø6:   END
[RZ1]  2Ø7: END (* FOR i *)
[242]  2Ø8: END; (* play *)
[ØØØ]  2Ø9:
[KNØ]  21Ø: END.
```

Listing 5. Die Unit »music.pas« läßt den Computer nach Ihren Noten spielen

Name	musik.pas
Sprache	Turbo Pascal ab Version 4.Ø
Grafikkarte	jede
Besonderheiten	keine

```
[V35]    1: { Programmname: MUSIK.PAS    }
[V94]    2: { (C)  Everts & Hagedorn     }
[ZB4]    3: {      DOS International     }
[C75]    4: { Autor: Hermann Rechberger  }
[ØØØ]    5:
[NT4]    6: (************ MUSIK.PAS ***************
[MK5]    7:  Kleines Demonstrationsprogramm
```

```
[CN6]     8:  fÜr die Anwendung der 'PLAY'-Anweisung
[ØØØ]     9:
[CD4]    1Ø:  VAR notenstring: STRING;
[9K4]    11:  (oder eine Stringkonstante)
[FA1]    12:  Aufruf:
[Ø63]    13:  play(notenstring);
[734]    14: ************************************)
[ØØØ]    15:
[9K2]    16: PROGRAM MUSIK;
[XK3]    17: {$R-,S-,I-,D-,F-,B-,V-}
[ØØØ]    18:
[SN2]    19: USES crt, music;
[ØØØ]    2Ø:
[6XØ]    21: CONST
[G65]    22:   Lieder: ARRAY[1..6] OF STRING =
[ØØØ]    23:
[FY4]    24:   (* kleine einstimmige Melodie *)
[6E5]    25:   ('O -2 L58Ø t s1  C E G|c2 c2|c2 c2|'+
[BP5]    26:    'c2 G2|E C E G c2 c2|c2 d c|H2 G2|'+
[GH5]    27:    '|F <H d f|h2 h2|h2 h2|h2 g2'+
[2A5]    28:    '|f g a h|>c2 c2|c2 A2|c4  s',
[ØØØ]    29:
[EN4]    3Ø:   (* P. Tschaikowsky: Motiv aus
[HØ5]    31:    dem italienischen Capriccio *)
[8K5]    32:    'O + L8ØØ T3 CE|EG2 CE|EG DF CE|' +
[G76]    33:    '<Gd2 >CE|DF2 DF+|EG CE2|P2>EC|' +
[8C6]    34:    'EG CE2<|P2 CE|EG2 CE|EG DF CE|' +
[H77]    35:    '<Gd2>CE|DF2 DF+|EG CE2|P2>EC|EG CE2<|'+
[3L5]    36:    'S1 P2 C|FA2 FA|FA GH Ac|EG2 EG|' +
[466]    37:    'EG FA GH|DF2 DF|DF EG FA|EC2 P1|' +
[UØ6]    38:    'C C C|FA2 FA|FA GH Ac|EG2 EG|' +
[NB5]    39:    'EG FA GH<gh fa eg|df ce Gd|cE4',
[ØØØ]    4Ø:
[YE4]    41:   (* Motiv aus der Film-Musik:
[RX2]    42:      Doktor Schiwago *)
[6N5]    43:    'O S +2 L8ØØ T3|CE3 GE2 Gd+|ce5 P1|'+
[3L4]    44:    'GH Hd Ac EG2 D+F+|FD5 P1|' +
[LV3]    45:    'FD3 GE2 FA  |GH5 P1|'+
[YH4]    46:    'FA EG D+F+ EG2 fA|Ge4 P2|'+
[RF3]    47:    'CE3 GE2 Gd+|ce5 P1|'+
[SG4]    48:    'GH Hd Ac EG2 D+F+|FD5 P1|'+
[XK3]    49:    'FD3 GE2 FA|GH5 P1|'+
[3T3]    5Ø:    'FA EG D+F+ DF2 < GH|cE5',
[ØØØ]    51:
[1C4]    52:   (* Glory, glory halleluja *)
[TC6]    53:    'O S + L58Ø EG3 DF|CE EG Ec Gd|ce4|'+
[CA5]    54:    'Ec4|FA3 GH|Ac GH Ac FA|EG4|CE4',
[ØØØ]    55:
[S13]    56:   (* eine Explosion *)
[RC3]    57:    'O L6Ø f e d+ d c+ c H '      +
[RL3]    58:    'A+ A G+ G F+ F E D+ D C+ C ',
[ØØØ]    59:
[8P1]    6Ø:   (* Fanfare *)
[6D2]    61:    '+ L58Ø C E G Ec3');
[ØØØ]    62:
[DC5]    63:   Titel: ARRAY[1..6] OF STRING[6Ø] =
[ØØØ]    64:
[NX4]    65:      ('Kleine einstimmige Melodie',
[855]    66:       'P. Tschaikowsky: Motiv aus ' +
[9X4]    67:       'dem Italienischen Capriccio',
[4T4]    68:       'Motiv aus der Film-Musik : ' +
[413]    69:       'Doktor Schiwago',
[V34]    7Ø:       'Glory, Glory Halleluja',
[ET2]    71:       'Eine Explosion',
[A14]    72:       'und Schlußfanfare');
[ØØØ]    73:
[ZH2]    74: VAR i: INTEGER;
[ØØØ]    75:
[XV2]    76: PROCEDURE Taste;
[4X1]    77: VAR c: CHAR;
[MUØ]    78: BEGIN
[2Z1]    79:   c := readkey;
[Y91]    8Ø:   CASE c OF
[LD1]    81:     #27: halt;
[B92]    82:     #Ø : c := readkey
[4MØ]    83:   END
[ZNØ]    84: END;
[ØØØ]    85:
[MUØ]    86: BEGIN
[DB1]    87:   clrscr;
[AF1]    88:   writeln(
[YP6]    89:   'Weiter geht''s mit irgendeiner Taste');
[AF4]    9Ø:   writeln('(Beenden mit ESC)');
[XF1]    91:   writeln;
[621]    92:   Taste;
[P31]    93:   REPEAT
[XC3]    94:     FOR i := 1 TO 6 DO BEGIN
[F93]    95:       writeln(Titel[i]);
[ØS2]    96:       play(Lieder[i]);
[NWØ]    97:       Taste
[ZNØ]    98:     END;
[2E1]    99:     writeln
[312]   1ØØ:   UNTIL FALSE
[KNØ]   1Ø1: END.
```

Listing 6. »musik.pas« ist ein Musikdemonstrationsprogramm

Name	spiele.pas
Sprache	Turbo Pascal ab Version 4.Ø
Grafikkarte	jede
Besonderheiten	keine

```
[F45]     1: { Programmname: SPIELE.PAS   }
[V94]     2: { (C)  Everts & Hagedorn     }
[ZB4]     3: {       DOS International    }
[C75]     4: { Autor: Hermann Rechberger  }
[ØØØ]     5:
[6T2]     6: PROGRAM SPIELE;
[SN2]     7: USES crt, music;
[ØØØ]     8:
[KC2]     9: CONST max = 4ØØ;
[Z72]    1Ø:       n: INTEGER = Ø;
[ØØØ]    11:
[ZH2]    12: VAR i: INTEGER;
[4U6]    13:     notenfeld: ARRAY[1..max] OF STRING[79];
[CZ2]    14:     notendatei: text;
[ØØØ]    15:
[MUØ]    16: BEGIN
[HC4]    17:   IF paramcount > Ø THEN BEGIN
[F65]    18:     assign(notendatei, paramstr(1));
[W33]    19:     reset(notendatei);
[S84]    2Ø:     WHILE (NOT eof(notendatei))
[383]    21:       AND (n < max) DO BEGIN
[L51]    22:       inc(n);
[LF5]    23:       readln(notendatei, notenfeld[n])
[ZNØ]    24:     END;
[XF3]    25:     FOR i := 1 TO n DO BEGIN
[3M3]    26:       IF notenfeld[i, 1] <> '*'
[F24]    27:         THEN play(notenfeld[i])
[ZW4]    28:         ELSE writeln(notenfeld[i]);
[WY3]    29:       IF keypressed THEN halt
[4MØ]    3Ø:     END
[4MØ]    31:   END
[KNØ]    32: END.
```

Listing 7. »spiele.pas« präsentiert Ihnen den Inhalt spezieller Notendateien akustisch

Name	bach2
Sprache	---
Grafikkarte	jede
Besonderheiten	ASCII-Datei, die Noten und Steuerzeichen für das Programm »spiele.pas« zur Verfügung stellt

```
[DF5]     1: * J.S. BACH: PRÄLUDIUM NR 2 IN C-DUR
[YN5]     2: * AUS DEM WOHLTEMPERIERTEN KLAVIER
[M2Ø]     3: *
[XM1]     4: T16 L128Ø S3
[4X4]     5: O5 C O3 C d+ G d F d+ G c D+ d+ O3 G d F
[7W4]     6: d+ G O5 C O3 C d+ G d F d+ G c D+ c+ G c
[W64]     7: F d+ G g+ C f G+ e G f G+ c F f G+
[195]     8: e G f G+ g+ C f G+ e G f G+ c F f G+ e G f
[KT4]     9: G+ h C f G+ d+ G f G+ d F f G+ d+ G f G+
[K55]    1Ø: h C f G+ d+ G f G+ d F f G+ d+ G f G+ O5 C
[ZP3]    11: O3 C g D+ f D g D+ d+ G g D+ f
[RD3]    12: D g D+ O5 C O3 C g D+ f D g
[PB3]    13: D+ d+ G g D+ f D g D+ O5 D+
[PB3]    14: O3 C g+ c g A+ g+ c d+ G+ g+
[ZP4]    15: g A+ g+ c O5 D+ O3 C g+ c g A+ g+ c d+
[YY4]    16: G+ g+ c g A+ g+ c O5 D O3 C f+ A e G f+ A
[AF4]    17: d F+ f+ A e G f+ A O5 D O3 C f+ A e
[DV4]    18: G f+ A d F+ f+ A e G f+ A O5 D O2 A+ O G
[LV4]    19: O3 A+ f+ A g A+ d G g A+ f+ A g A+ O5 D
[D13]    2Ø: O2 A+ O G O3 A+ f+ A g A+
[SG2]    21: O d G g A+ f+ A g A+
[D25]    22: O5 C O2 A+ O E O3 G d F e G c E e G d F
[L25]    23: e G O5 C O2 A+ O E O3 G d F e G c E e G
[BU2]    24: d F e G O5 C O2 G+ O F
[KZ3]    25: O3 G+ e G f G+ c F f G+ e G f G+
[GC3]    26: O5 C O2 G+ O F O3 G+ e G f G
[SN4]    27: c F f G+ e G f G+ a+ O2 G+ O F O3 D d+
[MH4]    28: C f D d F f D d+ C f D a+ O2 G+ O F
[224]    29: O3 D d+ C f D d F f D d+ C f D a+
[2V2]    3Ø: O2 G O G O3 D+ f D g D+
[462]    31: d+ G g D+ f D g D+
[RY4]    32: a+ O2 G O G O3 D+ f D g D+ d+ G g D+ f D
```

Listing 8. Ein Beispiel einer Notendatei für das Programm »spiele.pas«

```
[VZ1]   33: g D+ g+ C g D+ f
[273]   34: D g D+ d+ G+ g D+ f D g D+
[ØF2]   35: g+ C g D+ f D g D+ d+
[374]   36: G+ g D+ f D g D+ g+ D d F c D+ d F
[UF3]   37: A+ G+ d F c D+ d F g+ D d F c
[AV3]   38: D+ d F A+ G+ d F c D+ d F g D+ A+
[W24]   39: G G+ F A+ G d+ G+ A+ G G+ G A+ G g
[7K4]   4Ø: D+ A+ G G+ F A+ G d+ G+ A+ G G+ G A+ G
[AT3]   41: f D+ c A A+ G c G A F c A A+ G
[GM3]   42: c A f D+ c A A+ G c G A F c A
[V73]   43: A+ G c A f D d F c D+ d F H
[FY3]   44: G+ d F c D+ d F f D d F c D+ d F
[EF3]   45: H G+ d F c D+ d F f C d F c
[GN3]   46: E d F H G d F c E d F f C d
[5G3]   47: F c E d F H G+ d F c E d F d+
[GY3]   48: C c D+ H D c D+ G F c D+ H D c D+
[ØT3]   49: d+ C c D+ H D c D+ G F c D+ H D
[YC3]   5Ø: c D+ F O2 G+ d+ O3 C d O2 H
[WH2]   51: O D+ O3 C f O3 F d C
[2K3]   52: d O2 H O D+ O3 C F O2 G+ O D+
[913]   53: O3 C d O2 H O D+ O3 C f
[Y25]   54: F d+ C d O2 H O D+ O3 C F+ O2 A O C O3 D+
[HV4]   55: H D c D+ d+ C c D+ H D c D+ F+ O2 A O C
[4V4]   56: O3 D+ H D c D+ d+ C c D+ H D c D+ d+ O2
[D45]   57: G O C O3 D+ H D c D+ G F c D+ H D c D+ d+
[2Y4]   58: O2 G O C O3 D+ H D c D+ G F c D+ H D c D+
[TV4]   59: f+ O2 G O C O3 D+ H D c D+ A C c D+ H D
[AX4]   6Ø: c D+ f+ O2 G O C O3 D+ H D c D+ A C c D+
[1H4]   61: H D c D+ g O2 G O C O3 D+ H D c D+ d
[8X4]   62: F c D+ H D c D+ g O2 G O C O3 D+ H D c D+
[3S4]   63: d F c D+ H D c D+ g+ O2 G O C O3 D+ H
[VC4]   64: D c D+ d F c D+ H D c D+ g+ O2 G O C
[3L4]   65: O3 D+ H D c D+ d F c D+ H D c D+ O2 G
[954]   66: G H H O3 D D F F G+ G+ F F E E F F
[AS3]   67: H H F F d d H H G+ G+ F F E E
[S54]   68: F F O2 G G O3 C C D+ D+ G G c c G
[HW3]   69: F+ F+ G G d+ d+ c c g g d+ d+ c
[V44]   7Ø: c G+ G+ G G G+ G+ G G A A F+ F+ c
[CM3]   71: d+ d+ c c H H c c f+ f+ c c a
[DP4]   72: a f+ f+ d+ d+ c c H H c c O2 G G O5 D
[ZV4]   73: O2 G O5 C O2 G O5 D O2 G O5 D+ O2 G O5
[ØS3]   74: C O2 G O H O2 G O5 C O2 G O A
[VX4]   75: O2 G O5 C O2 G O H O2 G O5 C O2 G O5 D
[A24]   76: O2 G O H O2 G O A O2 G O H O2 G
[HW4]   77: O G O2 G O H D A C H D O5 C O D+ A C O3
[AT3]   78: H g a c f+ A a c g H a c h d g H
[V25]   79: f+ A g H d G O5 G O3 H O5 F O3 A O5 G O3
[A95]   8Ø: H O5 G+ O C O5 F O3 G+ O5 D+ O3 G O5 F O3
[KX4]   81: G+ O5 D O3 F O5 F O3 A O5 D+ O3 G O5 F
[LM4]   82: O3 A O5 G O3 H O5 D+ O3 G O5 D O3 F
[2V4]   83: O5 D+ O3 G O5 C O3 D+ O5 D+ O G O5 D O
[8L3]   84: F O5 D+ O G O5 F O G+ O5 D O
[BX3]   85: F O5 C O D+ O5 D O F O H D O5 D
[WT3]   86: O F O5 C O D+ O5 D O F O5 D+ O
[Y34]   87:  G O5 C O D+ H O D O5 C O D+ O G
[7T3]   88: C O5 C O D+ H D O5 C O D+ G+ F
[NE3]   89: O5 F O D O5 D+ O C O5 F O D
[GP3]   9Ø: O G D+ O5 D+ O C O5 D O3 H O5
[MR3]   91:  D+ O C F D O5 D O3 H O5 C O3 A
[F64]   92: O5 D O3 H d+ c O5 C O D+ H D O5 C
[2F4]   93: O D+ G+ O3 F f D+ C F D g D+ d+ C D
[LU2]   94: H d+ C F D d H c A d H O
[4Ø4]   95: SØ E4 S2 E C D E F G G+ A+ O5 C O
[P44]   96: A+ G+ G T16 F T32 G E O SØ T8 F
[5U3]   97: S2 F G F E F G G+ G F D+ D D+ F
[9Z3]   98: D D+ F T16 O3 H O2 C H O3 D F G+
[1B4]   99: G F H F O D O3 F H G+ G F T8 O3 E
[8Y3]   1ØØ: c+ A+ G c G+ F G+ G A+ G E G+ F D
[MG4]   1Ø1: F T8 O3 E G E F D O2 H O3 D O2 C G
[W34]   1Ø2: O3 C D E G A+ G T16 O3 G+ O C F D
[D34]   1Ø3: G+ O5 C O H O5 C O G F D O E O3 C
[NZ3]   1Ø4: O2 C O E O3 C O2 C O E O3 C O2 C
[KN4]   1Ø5: O E O3 C O2 C O E O2 C O3 C e C O2 C
[RF4]   1Ø6: O E O3 C O2 C O E O3 C O2 C O CE16
[M2Ø]   1Ø7: *
[LR2]   1Ø8: ***** (DAS WAR'S) *****
```

Listing 8. Ein Beispiel einer Notendatei für das Programm »spiele.pas«

Name	_puts.asm
Sprache	MASM ab Version 5.Ø, TASM
Grafikkarte	jede
Besonderheiten	Objekt-Datei (»_puts.obj«) zur Einbindung in die Pascal-Unit »gettext.tpu«

```
[6A4]    1: ; Programmname: _PUTS.ASM
[4E3]    2: ; (C)  Everts & Hagedorn
[8G3]    3: ;      DOS International
[NM4]    4: ; Autor: Hermann Rechberger
```

```
[ØØØ]    5:
[LK4]    6: ;Direkte Bildschirmausgabe
[ØØØ]    7:
[AD3]    8: DATA SEGMENT PUBLIC
[KM6]    9: EXTRN TEXTATTR:  BYTE; von der Unit Crt
[HD7]   1Ø: EXTRN SCREENADR: WORD; von der Unit Gettext
[DG1]   11: DATA ENDS
[ØØØ]   12:
[X24]   13: CODE SEGMENT BYTE PUBLIC
[KK3]   14:       ASSUME CS:CODE,DS:DATA
[ØØØ]   15:
[EE2]   16: PUBLIC  PUTSTR
[ØØØ]   17:
[Z77]   18: ;PROCEDURE PutStr(x, y: BYTE; str: STRING);
[5U4]   19: ;-----------------------------------------
[N84]   2Ø: str EQU DWORD PTR [BP+Ø6]
[WV3]   21: y   EQU BYTE  PTR [BP+1Ø]
[WV3]   22: x   EQU BYTE  PTR [BP+12]
[ØØØ]   23:
[KT2]   24: PUTSTR  PROC  FAR
[B91]   25:         PUSH  BP
[8W1]   26:         MOV   BP,SP
[ØØØ]   27:
[CU6]   28: ;ES mit Bildschirm-Anfangsadresse laden
[UZ2]   29:         MOV   AX,SCREENADR
[1W1]   3Ø:         MOV   ES,AX
[ØØØ]   31:
[2V2]   32:         MOV   BH,TEXTATTR
[P91]   33:         PUSH  DS
[MC5]   34:         LDS   SI,str  ;DS:SI zeigt auf str
[UT1]   35:         XOR   CH,CH
[Y65]   36:         MOV   CL,[SI] ;CX := Length(Str)
[2M4]   37: ;falls Nullstring, abbrechen
[142]   38:         JCXZ  RETURN
[ØØØ]   39:
[XP6]   4Ø: ;Berechnung des Bildschirm-Offsets in DI
[1U1]   41:         XOR   DH,DH
[VK1]   42:         MOV   DL,x
[WV1]   43:         MOV   DI,DX
[W31]   44:         DEC   DI
[VN1]   45:         MOV   AX,8Ø
[PK1]   46:         MOV   BL,y
[R31]   47:         DEC   BL
[K93]   48:         MUL   BL      ; AX :=  y*8Ø
[XØ4]   49:         ADD   DI,AX   ; DX :=  y*8Ø + x
[944]   5Ø:         SHL   DI,1    ; DI := (y*8Ø + x)*2
[ØØØ]   51:
[F16]   52: ;TextAttr und Zeichen aus Str nach AX
[BB4]   53:         MOV   AH,BH  ; AH := TextAttr
[ULØ]   54:         CLD
[7M2]   55: AUSGABE:INC   SI
[FN3]   56:         MOV   AL,BYTE PTR [SI]
[ZYØ]   57:         STOSW
[N22]   58:         LOOP  AUSGABE
[ØØØ]   59:
[YM2]   6Ø: RETURN: POP   DS
[571]   61:         POP   BP
[331]   62:         RET   8
[M22]   63: PUTSTR  ENDP
[ØØØ]   64:
[ZP1]   65: CODE    ENDS
[4MØ]   66:         END
```

Listing 9. Die Assemblerroutine »__puts.asm« sorgt in der Prozedur »PutStr« für eine schnelle Ausgabe von Zeichenketten

```
[9G5]    1: 8Ø ØB ØØ Ø9 5F 5Ø 55 54 53 2E 41 53 4D B2 88 1F
[PD5]    2: ØØ ØØ ØØ 54 75 72 62 6F 2Ø 41 73 73 65 6D 62 6C
[6E5]    3: 65 72 2Ø 2Ø 56 65 72 73 69 6F 6E 2Ø 31 2E 3Ø BA
[6E5]    4: 88 11 ØØ 4Ø E9 5C 82 39 14 Ø9 5F 5Ø 55 54 53 2E
[RA5]    5: 41 53 4D 5Ø 88 Ø3 ØØ 4Ø E9 4C 96 Ø2 ØØ ØØ 68 96
[F65]    6: Ø6 ØØ Ø4 44 41 54 41 46 98 Ø7 ØØ 68 ØØ ØØ Ø2 Ø1
[8D5]    7: Ø1 F5 96 Ø6 ØØ Ø4 43 4F 44 45 45 98 Ø7 ØØ 28 3A
[Y95]    8: ØØ Ø3 Ø1 Ø1 FA 8C 16 ØØ Ø8 54 45 58 54 41 54 54
[YA5]    9: 52 ØØ Ø9 53 43 52 45 45 4E 41 44 52 ØØ 36 9Ø ØD
[N55]   1Ø: ØØ ØØ Ø2 Ø6 5Ø 55 54 53 54 52 ØØ ØØ ØØ 69 88 Ø4
[4B5]   11: ØØ 4Ø A2 Ø1 91 AØ 3E ØØ Ø2 ØØ ØØ 55 8B EC A1 ØØ
[3N5]   12: ØØ 8E CØ 8A 3E ØØ ØØ 1E C5 76 Ø6 32 ED 8A ØC E3
[SV5]   13: 1F 32 F6 8A 56 ØC 8B FA 4F B8 5Ø ØØ 8A 5E ØA FE
[WY5]   14: CB F6 E3 Ø3 F8 D1 E7 8A E7 FC 46 8A Ø4 AB E2 FA
[4K5]   15: 1F 5D CA Ø8 ØØ 3A 9C ØB ØØ C4 Ø4 Ø6 Ø1 Ø2 C4 ØA
[4X2]   16: Ø6 Ø1 Ø1 B2 8A Ø2 ØØ ØØ 74
```

Listing 10. Das Hexdump-Listing der Objektdatei »__puts.obj«

Name	gettext.pas
Sprache	Turbo Pascal ab Version 4.Ø
Grafikkarte	jede
Besonderheiten	keine

```
[6F5]    1: { Programmname: GETTEXT.PAS  }
[V94]    2: { (C)  Everts & Hagedorn     }
[ZB4]    3: {      DOS International     }
[C75]    4: { Autor: Hermann Rechberger  }
[ØØØ]    5:
[A92]    6: UNIT GETTEXT;
[ØT3]    7: {$r-,l-,s-,i-,d-,f-,b-,v-}
[ØØØ]    8:
[SK1]    9: INTERFACE
[ØØØ]   1Ø:
[6XØ]   11: CONST
[CK3]   12:   ScreenAdr: WORD = $b8ØØ;
[FØ2]   13:   linelen = 8Ø;
[STØ]   14: TYPE
[2P4]   15:   linetyp = STRING[linelen];
[ØØØ]   16:
[A37]   17: PROCEDURE PutStr(x, y: BYTE; str: STRING);
[687]   18: FUNCTION gotlines(filename: STRING): WORD;
[5G5]   19: FUNCTION Line(i: WORD): linetyp;
[9X4]   2Ø: PROCEDURE Cursor(size: BYTE);
[WB7]   21: FUNCTION Itos(i: INTEGER; w: BYTE): STRING;
[ØØØ]   22:
[ØØØ]   23:
[215]   24: IMPLEMENTATION USES dos, crt;
[4W4]   25: (* textattr der Unit CRT wird
[SD4]   26:    in _puts.asm verwendet     *)
[ØØØ]   27:
[6XØ]   28: CONST
[ØØ5]   29:   nlinemax = $4ØØØØ DIV (linelen + 1);
[ØØØ]   3Ø:
[RNØ]   31: VAR
[M85]   32:   ScreenMode: BYTE ABSOLUTE 64:73;
[WT2]   33:   Textdatei: TEXT;
[3W6]   34:   Zeiger: ARRAY[1..nlinemax] OF ^linetyp;
[ØØØ]   35:
[FW3]   36: (************************************)
[MV4]   37: PROCEDURE PutStr; EXTERNAL;
[512]   38: {$L _puts}
[FW3]   39: (************************************)
[ØØØ]   4Ø:
[687]   41: FUNCTION gotlines(filename: STRING): WORD;
[PC2]   42: VAR i, n: WORD;
[9L2]   43:     zeile: linetyp;
[MUØ]   44: BEGIN
[YN4]   45:   assign(Textdatei, filename);
[XYØ]   46:   n := Ø;
[GZ2]   47:   reset(Textdatei);
[W92]   48:   i := ioresult;
[9F6]   49:   IF i = Ø THEN WHILE NOT eof(textdatei) DO
[MUØ]   5Ø:   BEGIN
[935]   51:     fillchar(zeile[1], linelen, ' ');
[Ø64]   52:     readln(textdatei, zeile);
[954]   53:     zeile[Ø] := chr(linelen);
[L51]   54:     inc(n);
[ST2]   55:     new(Zeiger[n]);
[7E3]   56:     Zeiger[n]^ := zeile
[ZNØ]   57:   END;
[7Y2]   58:   close(Textdatei);
[W92]   59:   i := ioresult;
[S42]   6Ø:   gotlines := n
[ZNØ]   61: END;
[ØØØ]   62:
[5G5]   63: FUNCTION Line(i: WORD): linetyp;
[MUØ]   64: BEGIN
[973]   65:   line := Zeiger[i]^
[ZNØ]   66: END;
[ØØØ]   67:
[9X4]   68: PROCEDURE Cursor(size: BYTE);
[6N4]   69: (* Cursor(Ø) versteckt Cursor
[NS6]   7Ø:    Cursor(2) erzeugt normalen Strich-Cursor
[1Z5]   71:    Cursor(12) erzeugt Block-Cursor       *)
[TU2]   72: VAR r: registers;
[MUØ]   73: BEGIN
[7G2]   74:   WITH r DO BEGIN
[DZØ]   75:     ah := 1;
[B31]   76:     cl := 13;
[S12]   77:     ch := 14 - size
[ZNØ]   78:   END;
[ZP1]   79:   intr(16, r)
[ZNØ]   8Ø: END;
[ØØØ]   81:
[WB7]   82: FUNCTION Itos(i: INTEGER; w: BYTE): STRING;
[UX4]   83: (* Integer to String (Länge w) *)
[DA2]   84: VAR s: STRING;
[MUØ]   85: BEGIN
[KZ1]   86:   str(i:w, s);
[5E1]   87:   ItoS := s
[ZNØ]   88: END;
[ØØØ]   89:
[DE4]   9Ø: (****** Initialisierung ******)
[1E5]   91: (* Monochrom- oder Farb-Adapter  *)
[MUØ]   92: BEGIN
[G56]   93:  IF ScreenMode = 7 THEN ScreenAdr := $bØØØ
[KNØ]   94: END.
```

Listing 11. Die Unit »gettext.pas« enthält Routinen zum Lesen von Textdateien

Name	zeige.pas
Sprache	Turbo Pascal ab Version 4.Ø
Grafikkarte	jede
Besonderheiten	keine

```
[F15]    1: { Programmname: ZEIGE.PAS    }
[V94]    2: { (C)  Everts & Hagedorn     }
[ZB4]    3: {      DOS International     }
[C75]    4: { Autor: Hermann Rechberger  }
[ØØØ]    5:
[XF2]    6: PROGRAM ZEIGE;
[ØT3]    7: {$r-,l-,s-,i-,d-,f-,b-,v-}
[H13]    8: USES Crt, GetText;
[ØØØ]    9:
[YE2]   1Ø: CONST rows = 23;
[ØØØ]   11:
[5G5]   12: VAR i, zeile, m, n, nline: INTEGER;
[Z81]   13:     y: BYTE;
[GC1]   14:     ch: CHAR;
[ØØØ]   15:
[W15]   16: PROCEDURE incr(step: INTEGER);
[MUØ]   17: BEGIN
[FZ1]   18:   inc(i, step);
[FP2]   19:   IF i > n THEN i := n
[ZNØ]   2Ø: END;
[ØØØ]   21:
[ZØ5]   22: PROCEDURE decr(step: INTEGER);
[MUØ]   23: BEGIN
[9B4]   24:   IF i > step THEN dec(i, step)
[KL1]   25:              ELSE i := 1
[ZNØ]   26: END;
[ØØØ]   27:
[MUØ]   28: BEGIN
[HC4]   29:   IF paramcount > Ø THEN BEGIN
[B25]   3Ø:     nline := gotlines(paramstr(1));
[RK3]   31:     IF nline > Ø THEN BEGIN
[DB1]   32:       clrscr;
[7M1]   33:       cursor(Ø);
[PYØ]   34:       i := 1;
[WR2]   35:       n := nline - rows;
[ZS2]   36:       IF n < 2 THEN n := 1;
[P31]   37:       REPEAT
[PZØ]   38:         y := 1;
[AB4]   39:         FOR zeile := i TO i + rows DO
[ØP3]   4Ø:           IF y <= nline THEN BEGIN
[M74]   41:             PutStr(1, y, line(zeile));
[XYØ]   42:             inc(y)
[ZNØ]   43:           END;
[5Ø2]   44:         m := i + rows;
[T94]   45:         IF m > nline THEN m := nline;
[SB5]   46:         TextAttr := $7Ø; (* Statuszeile *)
[SY2]   47:         PutStr(1, rows + 2,
[WK2]   48:                ' ' + paramstr(1) +
[5A3]   49:                ' / Zeile' + ItoS(i, 4) +
[NU2]   5Ø:                ' -' + ItoS(m, 4) + ' ');
[WH2]   51:         TextAttr := $Ø7;
[152]   52:         ch := readkey;
[LM1]   53:         CASE ch OF
[VF3]   54:           #$47 : i := 1;      (* home   *)
[5H3]   55:           #$4f : IF nline - rows > Ø
[XD3]   56:                    THEN i := nline - rows
[ØX2]   57:                    ELSE i := 1; (* end *)
[UK3]   58:           #$48 : decr(1);     (* up     *)
[ØT3]   59:           #$5Ø : incr(1);     (* down   *)
[Y04]   6Ø:           #$51 : incr(12);    (* pgdown *)
[BS3]   61:           #$49 : decr(12)     (* pgup   *)
[4MØ]   62:         END
[W54]   63:       UNTIL ch = #$1b;        (* Esc    *)
[DB1]   64:       clrscr;
[YX2]   65:     END ELSE writeln(
[WN3]   66:         'Datei nicht gefunden');
[DY5]   67:   END ELSE writeln('Dateiname fehlt');
[SE1]   68:   cursor(2)
[KNØ]   69: END.
```

Listing 12. Das Programm »zeige.pas« macht Ihnen den Einblick in Textdateien leicht

MS-DOS 4.0
Vom häßlichen Entlein zum stolzen Schwan

An MS-DOS kommt niemand vorbei, der einen PC bedienen will. Rund 35 Millionen Kopien sind heute weltweit im Einsatz. Damit ist MS-DOS zweifelsfrei das meistverbreitete und -kopierte PC-Programm aller Zeiten. Doch kann dieser Erfolg kaum darüber hinwegtäuschen, daß MS-DOS in der Vergangenheit das Image des häßlichen Entleins zugeschrieben wurde.

Die Sprache von MS-DOS war zweifellos kryptisch – kaum jemand verstand den Slang aus Prompts und unverständlichen Dateinamenfragmenten – seine Leistungsfähigkeit war auf 640 KByte Arbeitsspeicher und 32 MByte Festplattenspeicher begrenzt, und den beliebten Mäusen blieb der Zugang bislang völlig verwehrt. Allenfalls konnten Sie die tristesten Eigenheiten durch ein Mäntelchen wie Norton Commander, 1Dir und EMS-Software verbergen.

In der Version 4.0 wurde MS-DOS plötzlich »erwachsen«. Nie zuvor wurde so viel an einem Produkt geändert wie bei dem Versionssprung von MS-DOS 3.0 zu 4.0. Es präsentiert sich nun in völlig neuem Gewand: die »DOS-Shell«, Verwaltung großer Festplattenpartitionen, eine neue Dateiverwaltung, EMS-Unterstützung und ein komplett überarbeiteter Befehlssatz.

Die wohl wichtigste Neuerung in MS-DOS 4.0 ist die Benutzeroberfläche, die sogenannte »DOS-Shell«. Sie entspricht dem SAA-Standard (System-Anwendungs-Architektur) von IBM, einem Regelwerk, das ein einheitliches Erscheinungsbild für Software auf verschiedenen Rechnertypen bis hin zum Großrechner zum Ziel hat. Konkret ähnelt die DOS-Shell weitgehend dem OS/2-Presentation-Manager und der künftigen Version 3 von Windows. Es gibt Pulldown-Menüs, Dialogboxen, Bildlaufleisten und Icons und grafische Elemente, die Sie am einfachsten mit einer Maus bedienen. Weil grafische Darstellungsarten rechen- und speicherintensiv sind, bestimmen Sie, ob Sie im Text- oder im Grafikmodus mit der DOS-Shell arbeiten wollen oder ganz auf diese Oberfläche verzichten.

Die DOS-Shell hat den Vorteil, daß Sie sich keine Befehlssequenzen mehr merken müssen. Mit der Taste [F1] erhalten Sie jederzeit eine kontextbezogene Hilfefunktion. Ihre Anwendungen können Sie direkt aus der DOS-Shell heraus aufrufen. Hierzu steht das erweiterbare Hauptmenü »Programme Starten« zur Verfügung. Genau wie im Presentation Manager können Sie hier Ihre Programme mit Angaben zum Suchpfad und weiteren Hinweisen (etwa mit welcher Zeilenauflösung ein Programm gestartet werden soll) eintragen. Da Programme auch zu Gruppen zusammengefaßt werden können, lassen sich komplexe Hierarchien aufbauen. Mit Paßwörtern sichern Sie Ihre Programme.

MS-DOS verfügt jetzt endlich über eine ansehnliche Benutzeroberfläche und einen Dateimanager

Im Menü »Dateiverwaltung« lassen sich Festplatten- oder Diskettenstrukturen als Dateibäume anzeigen. Über Pulldown-Menüs stehen alle Verwaltungsbefehle wie »Kopieren«, »Löschen« oder »Umbenennen« zur Verfügung. Dateien lassen sich jedoch auch objektorientiert behandeln. Markieren Sie mit der Maus einen Dateinamen, so können Sie ihn direkt in ein anderes Verzeichnis verschieben. Sie können unter MS-DOS 4.0 auch Datendateien mit Programmen über Dateinamenerweiterungen verknüpfen. So legen Sie zum Beispiel fest, daß beim Aufruf einer beliebigen Datei mit der Endung »txt« automatisch Word gestartet und die markierte Datei geladen wird – Sie arbeiten »dokumentenorientiert«. Die Dateiverwaltung erlaubt aber noch mehr. So sehen Sie sich Dateien, ohne sie zu öffnen, im ASCII- oder Hexadezimal-Format an.

Bis zur Version 3.3 von MS-DOS durften Dateien und Plattenpartitionen nicht größer als 32 MByte sein. Die Version 4.0 unterstützt offiziell bis zu 512 MByte große Dateien und Partitionen. Die zur Version 4.0 gelieferten Gerätetreiber erlauben jedoch die Adressierung von zwei GByte großen Festplatten. Das theoretische Maximum liegt für MS-DOS 4.0 sogar bei zwei Terabyte.

Auch die 640-KByte-Arbeitsspeichergrenze ist gefallen. MS-DOS 4.0 unterstützt Expanded Memory nach den LIM-Spezifikationen (Lotus-Intel-Microsoft-Standard). Geeignete Anwendungsprogramme und Betriebssystemerweiterungen können nun einen Adreßbereich jenseits der 640-KByte-Grenze nutzen. Diesen Adreßbereich kann MS-DOS 4.0 aber auch zum Speichern von MS-DOS-Puffern und Fastopen-Daten verwenden, womit den Anwendungsprogrammen im Standardarbeitsspeicher mehr Speicherplatz bleibt. Die Befehle »fastopen« und »buffers« wurden wesentlich überarbeitet.

»fastopen« ist nun in der Lage, die Informationen über geöffnete Dateien im Cache zwischenzuspeichern. Mit »buffers« können mehr Datei-Puffer angelegt werden. Ein neuer Caching-Algorithmus vermindert die Zugriffszeit auf gepufferte Dateien. Alles in allem bedeutet dies eine höhere Leistungsfähigkeit gegenüber älteren MS-DOS-Versionen. Daneben verfügt MS-DOS 4.0 über einen sogenannten »Limulator«, der auf Rechnern mit 80386-Prozessor und Extended Memory EMS-Speicher simuliert.

Die meisten Betriebssystembefehle wurden in der Version 4.0 von MS-DOS verbessert beziehungsweise erweitert. Die wichtigsten Verbesserungen sind in dem Beitrag »Die Befehle von MS-DOS« zusammengefaßt. Einige dieser Verbesserungen waren sicherlich lange überfällig, aber sie sichern MS-DOS noch eine lange Zukunft. Die Speichergrenze von 640 KByte wird freilich wieder nur durch Tricks (EMS) umgangen. Die direkte Adressierbarkeit des Speicherplatzes jenseits der 640 KByte bleibt OS/2 und MS-DOS-Erweiterungen vorbehalten.

Multitasking, also das parallele Arbeiten mit mehreren Programmen, setzt ebenfalls Betriebssystemerweiterungen wie Windows voraus. Ein neues Dateisystem mit längeren Dateinamen, optimierter Verwaltung und damit besserer Leistung (vor allem im Datenbank- und Netzwerkbetrieb) bleibt für MS-DOS unerreichbar. Hierfür gibt es schließlich auch das kompatible OS/2 (ab Version 1.2).

Der Einstieg ins Computerleben wird mit MS-DOS 4.0 sicherlich leichter fallen. Die Notwendigkeit, auf alternative Oberflächen und Add-ons auszuweichen, wird sich für viele Einsteiger erst gar nicht mehr stellen – der Aufstieg zu Windows oder OS/2 ist dank der optischen Annäherung an diese Betriebssysteme gewiß unproblematischer geworden.

(Michael Kausch/as)

NEU

Wissen ist Macht.

Neu im DMV-Verlag: Computer-Wissen. Ein Magazin, das Ihnen Wissen über Ihren PC vermittelt, das andere nicht haben.

Unsere Themen: Alles, was Ihnen hilft, den PC besser zu nutzen und zu verstehen. Mit DMV-Computer-Wissen steht Ihnen das gesamte Know-how der DMV-Redaktionen in leichtverständlicher Form zur Verfügung.

Aus dem Inhalt:

Programme:
- Deutsche Fehlermeldungen in GW-BASIC
- Datei- und Verzeichniswahl mit Cursortasten
- BASIC-Programme automatisch strukturieren
- Konvertierungsprogramm von GW nach Turbo

Routinen:
- Umfangreiche Berechnungsfunktionen
- Konfiguration feststellen
- Hardcopy programmgesteuert
- Grafikroutinen
- Mausroutinen mit Testprogramm

Specials:
- Kalenderberechnungen
- Wahrscheinlichkeit und Statistik
- Literaturverzeichnis zum Thema BASIC

Und vieles anderes mehr:
Insgesamt über 500 Funktionen!

Damit Sie das Rad nicht ständig neu erfinden müssen, steht Ihnen im ersten Band von DMV-Computer-Wissen eine in Umfang und Vielfalt unvergleichliche Routinensammlung für Ihre eigenen Programme in GW-, Turbo- und Quick BASIC zur Verfügung. Alle Listings und Programme sind auch auf Diskette erhältlich.

AB SOFORT IM HANDEL

DMV-Computer-Wissen Band 1: Basic-Toolbox — **18,– DM***

DMV-Computer-Wissen I DATABOX (2 Stck. 5 1/4" und 1 Stck. 3 1/2") — **je 35,– DM***

* Unabhängig von der Anzahl der bestellten Artikel berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Die Datei »autoexec.bat« glasklar

MS-DOS 4.01 legt bei der Installation selbständig eine Datei namens »autoexec.bat« im Hauptverzeichnis an. Die einzelnen Befehle, die das Installationsprogramm dabei verwendet, erklärt es jedoch nicht. Darum übernimmt der folgende Beitrag diese Aufgabe. Sie erfahren hier, welche Befehle warum in der Datei »autoexec.bat« stehen, und was sie im einzelnen bewirken.

Zweifelsohne ist es sehr bequem, wenn Sie die Stapeldatei »autoexec.bat«, die beim Booten jedesmal automatisch ausgeführt wird, nicht selbst schreiben müssen. Es bringt aber auch einige Nachteile mit sich. Es werden nämlich einige MS-DOS-Programme installiert, die für Sie möglicherweise nicht erforderlich sind. Andere, die Sie aber benötigen würden, fehlen – je nachdem, was Sie mit Ihrem PC vorhaben. Damit Sie beurteilen können, wie Sie die »autoexec.bat« verändern müssen, sehen Sie sich die Datei erst einmal an (Bild).

Echo

Dieser Befehl hat mehrere Funktionen. Zum einen geben Sie mit

```
echo text
```

einen Text auf dem Bildschirm aus. Geben Sie

```
echo off
```

ein, so wird die Bildschirmausgabe unterdrückt, das heißt, alles, was der PC Ihnen normalerweise mitteilt, verschweigt er nun. In einer Stapeldatei angewandt, bewirkt das, daß Sie nicht sehen, welche Befehle im einzelnen ausgeführt werden. Nur die Bildschirmausgaben der aufgerufenen Programme sind zu sehen. Um den Befehl wieder rückgängig zu machen, gibt es den Parameter »on«. Sie geben also

```
echo on
```

ein, damit Sie wieder sehen, was vor sich geht. Wenn Sie übrigens in einer Zeile nur das Wort »on« oder das Wort »off« ausgeben möchten, verwenden Sie »echo.on« beziehungsweise »echo.off«. Der Punkt steht für ein unsichtbares Zeichen. Zur Darstellung einer Leerzeile auf dem Monitor schreiben Sie »echo.«. Damit können Sie aber wiederum keinen Punkt ausgeben, dafür geben Sie »echo..« ein. Der erste Punkt nach »echo« sorgt nur für ein unsichtbares Zeichen.
Geben Sie nur »echo« ein, teilt Ihnen MS-DOS mit, ob »echo« aktiviert ist (on) oder nicht (off). MS-DOS 4.0 verwendet außerdem das Zeichen »@« vor dem Befehl. Dieser »Klammeraffe« bewirkt, daß der nachfolgende Befehl bei der Ausführung nicht auf dem Bildschirm erscheint. Das ist beispielsweise dann eine praktische Sache, wenn Sie eine Stapeldatei schreiben möchten, die keinerlei Bildschirmausgaben produzieren soll. Das ist dann wichtig, wenn auf dem Bildschirm noch Informationen stehen, die Sie benötigen, nachdem Sie die Stapeldatei ausgeführt haben, oder wenn Sie das Ergebnis der verschiedenen Befehle der Stapeldatei in eine Textdatei umleiten möchten. Dafür schreiben Sie in die erste Zeile »echo off«. Allerdings ist »echo« noch nicht »off«, wenn diese Zeile ausgeführt wird. »echo off« würde also noch auf dem Bildschirm stehen. Damit auch das vermieden wird, setzen sie den Klammeraffen direkt vor den Befehl.

Set Comspec...

Bei Programmen, die einen hohen Speicherbedarf haben, wird »command.com« aus dem Arbeitsspeicher verbannt und nach Beendigung des Programms erneut geladen. Damit MS-DOS auch weiß, wo es die Datei »command.com« findet, benutzt es die vorbelegte Umgebungsvariable »comspec«. In dieser Variable steht der Pfad, unter dem »command.com« abgelegt ist.
Sie können mit Hilfe des Kommandos

```
shell=c:\dos\command.com
```

in der Datei »config.sys« dafür sorgen, daß beim Hochfahren des PC der Kommando-Interpreter nicht aus dem Hauptverzeichnis geladen wird, sondern aus dem Unterverzeichnis »dos«. Dafür muß in der Datei »autoexec.bat« die Zeile

```
set comspec=c:\dos\command.com
```

stehen. Dann können Sie die Datei »command.com« aus dem Hauptverzeichnis löschen und haben somit Platz für ein weiteres Unterverzeichnis.
Die übliche Lösung besteht aber darin, »command.com« im Hauptverzeichnis zu belassen und »comspec« auf »c:\command.com« zu setzen. Es gibt nämlich Programme, die auch »command.com« benutzen, aber danach nur im Hauptverzeichnis suchen. Dann stürzt das System natürlich ab, wenn der Kommando-Interpreter nicht im Hauptverzeichnis steht. Am besten, Sie ersetzen also die Zeile

```
set comspec=c:\dos\command.com
```

durch

```
set comspec=c:\command.com
```

Damit sind Sie und Ihr PC für alle Zweifelsfälle gerüstet.

Verify

MS-DOS bietet Ihnen die Gelegenheit, Schreibvorgänge auf Diskette oder Festplatte zu überprüfen. Dabei wird sichergestellt, daß die geschriebenen Daten auch tatsächlich korrekt sind. Ob diese Überprüfung stattfinden soll oder nicht, stellen Sie mit dem Befehl »verify« ein. Entweder verwenden Sie

```
verify on
```

um alle Schreibvorgänge zu überprüfen oder

```
verify off
```

um die Überprüfung zu unterbinden. Das hat den Vorteil, daß Dateioperationen schneller vonstatten gehen.

```
@echo off
set comspec=c:\dos\command.com
verify off
path c:\dos
append /e
append c:\dos
prompt $p$g
ver
keyb gr,,c:\dos\keyboard.sys
dosshell
```

So sieht die von MS-DOS 4.01 erzeugte Stapeldatei aus

Treten auf Ihrer Festplatte häufiger Schreibfehler auf, hat es Sinn, die Überprüfung einzuschalten. In der Regel ist das aber nicht notwendig. Beim Befehl »copy« können Sie immer noch den Parameter »/v« angeben, der die Überprüfung für diesen Schreibvorgang aktiviert.

Path

Sobald Sie ein Kommando in der MS-DOS-Ebene eingeben, sucht MS-DOS im Kommando-Interpreter, ob dieser Befehl zu denjenigen gehört, die sich im Arbeitsspeicher befinden. Ist dem so, wird er ausgeführt. Andernfalls beginnt MS-DOS, im aktuellen Verzeichnis danach zu suchen. Wird der Befehl auch dort nicht vorgefunden, erhalten Sie die Fehlermeldung »Falscher Befehl oder Dateiname«. Damit Programme auch mit Dateien arbeiten können, die sich nicht im gleichen Verzeichnis befinden, gibt es die Umgebungsvariable »path«. Durch diese Variable sieht MS-DOS nach, in welchen Verzeichnissen, außer dem aktuellen, es noch nach dem eingegebenen Befehl suchen soll. Sie möchten zum Beispiel im Unterverzeichnis »docs« eine Datei »fgrep.doc« ansehen. Damit das etwas bequemer vor sich geht, soll die Datei seitenweise auf dem Monitor erscheinen. Dazu müßten Sie eigentlich

```
\dos\more < fgep.doc
```

eingeben. Wenn Sie aber die Zeile

```
path c:\dos
```

in Ihrer »autoexec.bat« eingetragen haben, genügt

```
more < fgrep.doc
```

Dann sucht MS-DOS nämlich auch im Verzeichnis »dos« nach dem eingegebenen Befehl. Sie sehen, das erspart Ihnen eine Menge Tipparbeit. Diese Zeile erweitern Sie noch auf verschiedene Verzeichnisse, in denen sich Programme befinden, auf die Sie aus mehreren Unterverzeichnissen zugreifen möchten. Die einzelnen Unterverzeichnisse trennen Sie durch Strichpunkte ab. Zum Beispiel so:

```
path c:\dos;c:\bat;c:\tools
```

Append

Mit Hilfe dieses Befehls veranlassen Sie MS-DOS, auch Dateien, die nicht ausführbar sind, in mehreren Unterverzeichnissen zu suchen. Das ist zum Beispiel dann nützlich, wenn Sie mit Programmen arbeiten, die auf Daten in mehreren Verzeichnissen zugreifen sollen, aber nicht die Möglichkeit haben, diese Pfade vorher anzugeben. Dazu verwenden Sie die Zeile

```
append /x
```

Allerdings machen viele Programme nicht von den MS-DOS- Funktionen Gebrauch, weil diese sehr langsam sind – dann hilft auch append nichts. Aber sämtliche MS-DOS-Befehle unterstützen append. So könnten Sie beispielsweise mit »type« Dateien anzeigen, die sich nicht im aktuellen Verzeichnis befinden, ohne den Pfad beim Aufruf anzugeben. Für die MS-DOS-Befehle genügt aber der Parameter »/e«.

»/x« dient lediglich dazu, die mit »append« angegebenen Verzeichnisse auch für andere Programme nutzbar zu machen, die aber, wie gesagt, die MS-DOS-Funktionen verwenden müssen.

Um die Verzeichnisse anzugeben, in denen MS-DOS nach den Daten suchen soll, verwenden Sie ebenfalls append, und zwar wie folgt:

```
append c:\dos
```

Das funktioniert also genauso wie bei »path«.

Prompt

Das Kommando »prompt« ist immer dann wichtig, wenn Sie direkt in der MS-DOS-Kommandoebene arbeiten. Sobald der PC eine Eingabe von Ihnen erwartet, erscheint das sogenannte »Prompt«. Wenn Sie nichts anderes vorgegeben haben, lautet es »C>«. Weisen Sie aber in der Autoexec-Datei »prompt pg« an, so zeigt Ihnen MS-DOS immer das aktuelle Verzeichnis. Dafür steht der Parameter »$p«, der das Größerzeichen bewirkt. Wenn Sie Zeichen ohne das Dollar-Symbol eingeben, erscheinen diese »wörtlich« im Prompt. Beispielsweise führt

```
prompt Das aktuelle Verzeichnis ist
$p...
```

zu dem Ergebnis: »Das aktuelle Verzeichnis ist C:\...«. Welche Parameter Ihnen außer »$p«, »$g« und beliebigem Text noch zur Auswahl bereitstehen, entnehmen Sie der Tabelle.

Parameter	Resultat im Prompt
$$	das Zeichen »$«
$_	Wagenrücklauf + Zeilenvorschub
$b	das Zeichen »|«
$d	aktuelles Datum
$e	[Esc] (ASCII-Code 27)
$g	das Zeichen »>«
$h	Backspace (löscht das vorhergehende Zeichen)
$l	das Zeichen »<«
$n	aktuelles Laufwerk
$p	aktuelles Unterverzeichnis (mit Laufwerksbuchstaben)
$q	das Zeichen »=«
$t	aktuelle Uhrzeit
$v	verwendete MS-DOS-Version

Die Tabelle zeigt die Parameter des Befehls »prompt«

Ver

Wenn Sie zu den Menschen gehören, die pausenlos zwischen verschiedenen MS-DOS-Versionen hin- und herwechseln, dann interessiert Sie sicher der Ver-Befehl. Er zeigt die momentan geladene Version von MS-DOS an. Normalerweise ist dieses Kommando allerdings – besonders in der Datei »autoexec.bat« – vollkommen überflüssig.

Keyb

Je nach Sprache sind die Tasten der PC-Tastatur unterschiedlich belegt. Damit der PC weiß, daß Sie eine deutsche Tastatur benutzen, müssen Sie ihn davon erst einmal in Kenntnis setzen. Das bewirken Sie mit dem Befehl »keyb«. »keyb« ist ein Tastaturtreiber, der die Codes der Tastatur den entsprechenden Zeichen zuordnet. Damit er dem jeweiligen Code das entsprechende deutsche Zeichen zuordnen kann, geben Sie den Parameter »gr« an. Die Zeichensätze, die dann auf dem Bildschirm angezeigt werden, stehen in »Codeseiten«. Die Nummer der Codeseite, die Sie verwenden möchten, geben Sie nach dem Länderkennzeichen »gr« durch ein Komma getrennt an. Für Deutschland ist das der Wert 437. Wieder durch ein Komma getrennt, beschreiben Sie den Pfad für die Tastaturdefinitionstabelle, die normalerweise »keyboard.sys« lädt. Daraus ergibt sich die Zeile

```
keyb gr, 437, c:\dos\keyboard.sys
```

Sie müssen allerdings nicht mit Codeseiten arbeiten. Dann lautet das Kommando

```
keyb gr,,c:\keyboard.sys
```

wie Sie es auch in der automatisch erzeugten Datei »autoexec.bat« finden.

Dosshell

Die Dosshell ist die grafische Benutzeroberfläche von MS-DOS 4.01. Sie erspart viel Tipparbeit, weil sie sich über Menüs steuern läßt. So können Sie beispielsweise Programme mit der Maus starten oder Dateien mit Hilfe der Cursor-Tasten auswählen. Die Dateien werden von der Dosshell in einem Fenster angezeigt. Die Benutzeroberfläche starten Sie ganz einfach mit dem Kommando

```
dosshell
```

Wenn Sie dennoch einen Befehl in der regulären MS-DOS-Umgebung starten möchten, können Sie das unter einem Menüpunkt der »Dosshell« ausführen.

(bk)

MS-DOS 4.01: Was ist neu in »config.sys«?

Die Konfigurationsdatei »config.sys« ist jedem PC-Benutzer bestens bekannt. Die wichtigsten Systemparameter werden hier festgelegt, und jeder Anwender hat sich im Laufe der Zeit seine »ganz persönliche« Konfiguration erarbeitet. Mit der Version 4.01 des MS-DOS-Betriebssystems eröffnen sich auf diesem Gebiet einige neue Möglichkeiten.

Viele Rechner, vor allem die mit einer großen Festplatte ab 100 MByte, werden heute schon »ab Werk« mit der MS-DOS-Version 4.01 ausgeliefert. Gegenüber MS-DOS 3.3 haben die Programmierer von Microsoft einige neue Befehle implementiert, interessant sind vor allem die Neuerungen in der Steuerungsdatei »config.sys«.

Schon bei der Installation der Betriebssystemversion 4.01 sehen Sie, daß sich an MS-DOS einiges geändert hat. Die Anpassung von MS-DOS an Ihre individuelle Rechnerkonfiguration erfolgt über das Programm »select«, das im Dialog mit dem Benutzer die Datei »config.sys« einrichtet (»select« wird automatisch ausgeführt, wenn Sie den Rechner von der ersten MS-DOS-4.01-Systemdiskette starten). Außer dem Kopieren diverser Hilfsprogramme auf die Festplatte erstellt »select« aber unter anderem auch eine Datei mit dem Namen »config.400«, falls bereits ein »config.sys« im Hauptverzeichnis von Laufwerk C vorliegt.

Für den Fall, daß Sie MS-DOS zum ersten Mal auf einem Computer installieren, erhält »config.400« gleich den Namen »config.sys«. MS-DOS 4.01 löscht also eine bereits bestehende »config.sys« nicht, damit dort enthaltene, wichtige Informationen nicht verlorengehen können. Auf jeden Fall sollten Sie sich »config.400« einmal genauer ansehen, um eventuell einige Zeilen in Ihre »config.sys« zu übernehmen. Bei den meisten Installationen wird »config.sys« beziehungsweise »config.400« so aussehen, wie Bild 1 zeigt. Viele der darin enthaltenen Befehle kennen Sie – zumindest teilweise – bereits aus früheren MS-DOS-Versionen.

Der erste Befehl lädt den Treiber »country.sys« aus dem MS-DOS-Unterverzeichnis »c:\dos« und installiert ihn mit der deutschen Länderkennung »49«. Dies bewirkt, daß MS-DOS zum Beispiel die Reihenfolge für die Datumsangabe des Dir-Befehls auf das in Deutschland übliche Format »Tag.Monat.Jahr« einstellt.

Zwischen den beiden Kommas des Country-Befehls können Sie optional eine dreistellige Zahl angeben, die eine sogenannte »Codepage« installiert. Eine Codepage ist eine Zeichensatztabelle, die landesspezifische Sonderzeichen mit Umlauten enthält. MS-DOS stellt in der Version 4.01 eine ganze Reihe verschiedener, meist mehrsprachiger Codepages zur Verfügung. Mehr zu diesem Thema etwas später.

Die Anweisungen »buffers« und »files« legen, wie bereits bei älteren Versionen von MS-DOS, die Anzahl der »Puffer« für Plattenzugriffe und die maximale Zahl der gleichzeitig geöffneten Dateien fest. Wenn Sie beispielsweise zweimal hintereinander »dir« eingeben, dann greift das Betriebssystem nur beim ersten Mal auf die Festplatte zu. Beim zweiten Aufruf hat es alle benötigen Informationen noch in den Puffern gespeichert.

Mit »Lastdrive« definieren Sie, bis zu welchem Buchstaben MS-DOS Platz für die Laufwerkskennungen reservieren soll. Normalerweise benötigen Sie diesen Befehl nur, wenn Sie nach dem Booten »virtuelle«, also nicht physikalisch, sondern nur scheinbar vorhandene Laufwerke mit »subst«, »join«, »assign« oder einer Netzwerk-Software einrichten möchten. Für RAM-Disks und spezielle Gerätetreiber, die von »config.sys« aus geladen werden, brauchen Sie die Laufwerksnamen nicht zu reservieren.

Die Anweisung »shell« definiert den Kommandoprozessor, der die Autoexec-Datei bearbeitet. In Bild 1 wird der normale Kommandoprozessor »command.com« aus dem Verzeichnis »c:\dos« installiert, und zwar mit einem Umgebungsbereich von 256 Byte. Im Umgebungsbereich speichert MS-DOS seine internen Parameter, wie zum Beispiel den Zugriffspfad »path« oder die Eingabeaufforderung »prompt«.

```
BREAK=ON
COUNTRY=49,,C:\DOS\COUNTRY.SYS
BUFFERS=20
FILES=8
LASTDRIVE=E
SHELL=C:\DOS\COMMAND.COM /P /E:256
DEVICE=C:\DOS\ANSI.SYS
DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,437,1)
DEVICE=C:\DOS\PRINTER.SYS LPT1=(4201,437,1)
INSTALL=C:\DOS\KEYB.COM US,,C:\DOS\KEYBOARD.SYS
INSTALL=C:\DOS\FASTOPEN.EXE C:=(50,25)
INSTALL=C:\DOS\NLSFUNC.EXE C:\DOS\COUNTRY.SYS
```

Bild 1. So sieht die Datei »config.sys« aus, wenn Sie MS-DOS 4.01 neu installiert haben

Die Codepages von MS-DOS 4.01 werden Sie normalerweise nie benötigen

Die beiden folgenden Device-Anweisungen, die jeweils Treiberprogramme installieren, können Sie bei jedem normalen IBM-kompatiblen Rechner ohne Bedenken löschen. Der Treiber »display.sys« für den Bildschirm wird mit Codepage 437 (eine US-amerikanische Zeichensatztabelle, die auch deutsche Umlaute enthält) im EGA-Modus initialisiert, und der Drucker wird als IBM-Drucker »4201« – ebenfalls mit Codepage 437 – eingestellt. Alle Standardprogramme für Drucker- und Grafikanwendungen greifen jedoch über ihre eigenen Routinen auf die Geräte zu und die Treiber »display.sys« und »printer.sys« verbrauchen dann nur überflüssigerweise kostbaren Arbeitsspeicher.

Wirklich interessant wird es in der neuen Version von »config.sys« bei der Anweisung »install«. Sie ist die erste bedeutende Änderung gegenüber allen MS-DOS-Versionen vor 4.01. Mit ihrer Hilfe lassen sich speicherresidente Programme schon während der Ausführung von »config.sys« installieren, also noch bevor die Stapeldatei »autoexec.bat« bearbeitet wird. Im Lieferumfang von MS-DOS 4.01 befinden sich insgesamt vier Programme, die sich auf diese Art aktivieren lassen, nämlich »keyb.com«, »nlsfunc.exe«, »fastopen.exe« und »share.exe«.

In Bild 1 wird der Tastaturtreiber »keyb.com« mit der amerikanischen Tastenbelegung aufgerufen, und zwar obwohl die Installation mit einer deutschen MS-DOS-Version er-

folgte. Diesen Fehler korrigieren Sie mit einem Editor, indem Sie die Zeichen »US« durch »GR« (»german«, »deutsch«) ersetzen. Zwischen den beiden Kommas können Sie, wie bei »country« auch, die Nummer einer Codepage einfügen, die beim Systemstart für die Tastatur verwendet werden wird. Am Ende der Zeile müssen Sie noch angeben, in welchem Verzeichnis sich die Datei »keyboard.sys« befindet. Diese Datei enthält die zum Landeskürzel zugehörigen, länderspezifischen Informationen für die Tastatur. In der vom Installationsprogramm vorgegebenen Autoexec-Datei wird der Tastaturtreiber übrigens noch einmal installiert, obwohl er ja bereits durch »config.sys« eingerichtet wurde. Den zweiten Aufruf von »keyb.com« in »autoexec.bat« können Sie also getrost löschen. Im Moment ist es noch unklar, welche Vorteile die Installation eines Programms über den Install-Befehl in der Datei »config.sys« bringt. Manche Programmierer sind der Meinung, daß sich das Programm dadurch »harmonischer« in die MS-DOS-Umgebung einfügt, weil es noch vor dem Kommandointerpreter in den Speicher geladen wird – für den Normal-Anwender erkennbare Vorteile zeigen sich allerdings nicht.

Die Programmierer streiten sich noch, ob der Install-Befehl in »config.sys« konkrete Vorteile bringt

Der zweite Install-Befehl aktiviert das Programm »fastopen.exe«, welches einen beschleunigten Zugriff auf Massenspeicher ermöglicht. Dies erreicht MS-DOS, indem es die Positionen der zuletzt benötigten Dateien speichert und bei einem erneuten Zugriff somit die Zeit zum Auffinden dieser Daten entfällt.
In Bild 1 wird »fastopen« für das Laufwerk C so installiert, daß insgesamt 50 Speicherplätze für Dateien und 25 Speicherplätze für die zuletzt gelesenen Daten bereitgestellt werden (sogenanntes »caching«). Wenn Sie häufig mit Programmen arbeiten, die direkt auf die Festplatte zugreifen, wie es beispielsweise bei Disk-Optimizern und vielen Cache-Programmen der Fall ist, dann sollten Sie die Fastopen-Installation unbedingt aus der Config-Datei entfernen. Sonst könnte es einmal passieren, daß die Informationen, die Fastopen über die Position der Dateien auf der Platte gespeichert hat, nicht mit der tatsächlichen Anordung übereinstimmen. Die Folgen können verheerend sein, wenn nicht rechtzeitig ein Neustart (»Reboot«) des Systems erfolgt und Fastopen somit die Daten neu liest.
Die letzte Anweisung des Beispiels in Bild 1 zeigt eine weitere Neuerung von MS-DOS 4.01, nämlich die nun implementierte, umfangreiche Unterstützung der bereits angesprochenen Zeichensatztabellen (»codepages«). Um den Vorteil dieser flexiblen Anpassung an verschiedene landesspezifische Zeichensätze nutzen zu können, müssen Sie das Programm »nlsfunc.exe« ausführen, das ihren Computer auf die Verwendung der Codepages vorbereitet. Wollen Sie die verfügbaren Codeseiten häufig benutzen, so empfiehlt es sich, den Aufruf von »nlsfunc« mittels Install-Befehl gleich von »config.sys« aus vorzunehmen. Vergessen Sie nicht, wie in obigem Beispiel die Datei anzugeben, die die Informationen über die Codeseiten enthält. Normalerweise ist dies »country.sys«.
Falls Sie sich für die genaue Verwendung von »codepages« interessieren, sollten Sie sich näher mit der Anweisung »mode« befassen. Dieses Dienstprogramm erhielt in MS-DOS 4.01 ebenfalls einige neue Funktionen zur Vorbereitung und Einstellung von Codepages spendiert.
Eine letzte Neuerung bezüglich der in »config.sys« verfügbaren Anweisungen wird Ihnen vielleicht bekannt vorkommen, wenn Sie einmal in der Programmiersprache Basic programmiert haben. Auch in »config.sys« besteht nun die Möglichkeit, Kommentare mit einem vorangestellten Rem-Befehl (»remark«, »Bemerkung«) einzufügen. Sie sollten von dieser Möglichkeit auch dann Gebrauch machen, wenn Ihre Config-Datei nur wenige Zeilen lang ist.
Beispielsweise können Sie sinnvollerweise einige »Rem-Zeilen« einfügen, die die Parameter eines Gerätetreibers kurz erläutern. Sollten Sie dann einmal eine Änderung an »config.sys« vornehmen, müssen Sie die entsprechenden Informationen nicht mehr im MS-DOS-Handbuch nachschlagen.
Eine mögliche Variante einer ausführlich und sinnvoll gestalteten »config.sys« sehen Sie in Bild 2. Erläuterungen zu den einzelnen Anweisungen finden Sie in den Rem-Zeilen.

(Thomas Rueß/tr)

```
REM Unterbrechungen mit ^C auch bei Laufwerkszugriffen erlauben
BREAK ON

REM Zahl der Pufferspeicher für Massenspeicherzugriffe festlegen
BUFFERS=20

REM Zahl der maximal gleichzeitig offenen Dateien festlegen
FILES = 30

REM Kommandoprozessor »command.com« installieren
REM »/P« erzwingt die Ausführung von »autoexec.bat«
REM »/E:1024« vergrößert den MS-DOS-Umgebungsbereich auf 1024 Byte
SHELL=C:\DOS\COMMAND.COM /P /E:1024

REM Umstellen des Systems auf deutsche Norm für Zeit, Datum etc.
REM »49« ist die Landeskennzahl »Bundesrepublik Deutschland«
REM »734« installiert die Standard-Codeseite (Zeichensatz)
COUNTRY=49,437,C:\DOS\COUNTRY.SYS

REM RAM-Disk anlegen
REM (256 KByte Größe, 512 Byte pro Sektor, max.64 Einträge im
Hauptverzeichnis)
REM »/E« installiert die RAM-Disk im Extended Memory (»AT-Speicher«)
REM »/A« hingegen würde bedeuten: RAM-Disk im Expanded Memory (EMS)
anlegen
DEVICE=C:\DOS\RAMDISK.SYS 256 512 64 /E

REM Verlagern des Video-BIOS in den RAM-Speicher
REM Der Treiber »rambios.sys« ist Grafikkarten-spezifisch und wird
meistens
REM zusammen mit einer EGA- oder VGA-Grafikkarte ausgeliefert
DEVICE=C:\DOS\RAMBIOS.SYS

REM Laden des deutschen Tastaturtreibers unter Verwendung der Codeseite
»437«
INSTALL=C:\DOS\KEYB.COM GR,437,C:\DOS\KEYBOARD.SYS
```

Bild 2. Für die meisten Rechner ist diese Version der Datei »config.sys« besser geeignet als die Vorgabe des MS-DOS-Installationsprogramms

Die neuen Befehle von MS-DOS 4.01

Wenn Sie von MS-DOS 3.3 auf die Version 4.01 umsteigen, dann interessiert Sie natürlich vorrangig, was sich denn jetzt alles geändert hat. Um alle Neuerungen des Betriebssystems kennenzulernen, ist es ein reichlich langwieriges Unterfangen, das ganze Handbuch durchzublättern. Wir stellen Ihnen deshalb kurz und bündig die neuen Befehle und neuen Optionen bereits bekannter MS-DOS-Befehle vor.

Die neue Version 4.01 des MS-DOS-Betriebssystems besitzt einige Fähigkeiten, die keine der Vorgängerversionen aufzuweisen hat. So sind einige neue Befehle hinzugekommen, und manches bekannte Systemprogramm erhielt ein neues Gesicht oder zusätzliche Optionen.
Das Handbuch der neuen Version 4.01 von MS-DOS verwirrt den Anwender wie schon bei früheren Versionen mit knappen und häufig unverständlichen Beschreibungen. Während es sich für den Einsteiger lohnt, spezielle Literatur zu kaufen – die es inzwischen auch für die neue Betriebssystemversion zahlreich gibt – und sich in die Materie einzulesen, interessiert den erfahreneren Anwender wohl mehr, welche Neuerungen denn nun konkret in MS-DOS 4.01 zu bewundern sind. Damit Sie sich die Arbeit sparen können, das gesamte Handbuch nach neuen Befehlen und Befehlsoptionen zu durchforsten, sind im folgenden alle diesbezüglichen Neuheiten zusammengestellt.
Die aufgeführten Befehle existieren zum Teil auch in älteren Versionen von MS-DOS. Die angegebene Syntax enthält alle möglichen Befehlsoptionen, der jeweils folgende Textteil geht aber nur auf neue Funktionen unter MS-DOS 4.01 ein.

append

Syntax: APPEND [<Pfad1>] [;<Pfad2> [...]] [/X[:OFF|ON]] [/PATH:ON|OFF] [/E] [;]

Der Befehl »append« erlaubt es, einen Suchpfad für Datendateien festzulegen, ähnlich der Path-Anweisung, die einen Suchweg für ausführbare Dateien mit der Endung »exe«, »com« oder »bat« definiert. »append« bezieht sich auf Dateien, die über die Standard-MS-DOS-Funktionen zum Öffnen einer Datei aktiviert werden.
Die Option »/x« oder »/x:on« bewirkt, daß sich »append« wie »path« verhält und in den angegebenen Verzeichnissen auch nach ausführbaren Programmdateien sucht. Der Append-Befehl ersetzt dann die Pfadangabe von MS-DOS. Der Parameter »/x:off« schaltet diese Funktion wieder aus, so daß »append« nur nach Dateien sucht, die nicht auf »exe«, »com« oder »bat« enden.
Die neue Option »/path:on« weist »append« an, in den angegebenen Verzeichnissen nach Datendateien unabhängig davon zu suchen, ob den Dateien eine Laufwerks- oder Pfadangabe vorangestellt ist. Eine Pfadangabe wird also von »append« ignoriert, und alle angegebenen Verzeichnisse werden nach der gewünschten Datei durchsucht. Wenn Sie beispielsweise »c:\dos\format.com« eingeben, MS-DOS aber im Unterverzeichnis »c:\dos« kein Programm namens »format.com« findet, dann durchsucht es auch die anderen Pfadverzeichnisse des Append-Befehls nach diesem Programm.
Um dies wieder rückgängig zu machen, verwenden Sie den Schalter »/path:off«. »append« sucht dann nur noch nach Datendateien, wenn diesen weder Laufwerksbuchstabe noch Suchpfad vorangestellt wurde.

backup

Syntax: BACKUP <Quelle> [<Dateiname(n)>] <Ziel> [/S] [/M] [/A] [/D:x] [/T:x] [/L:x] [/F:<Größe>]

Die bereits aus früheren MS-DOS-Versionen bekannte Option »/F«, die das Formatieren der Diskette im »Ziellaufwerk« erlaubt, ist um die Möglichkeit ergänzt worden, die Diskettenkapazität direkt vorwählen zu können. Sie eignet sich dann, wenn Sie unformatierte Disketten für das Backup verwenden möchten. Für den Parameter »Größe« setzen Sie einen der Werte 160, 180, 320, 360, 720, 1200 oder 1440 ein, je nachdem welche Disketten Sie verwenden möchten und welche Formate von Ihrem Computer unterstützt werden.

del und erase

Syntax: DEL [<Dateiname(n)>] [/P]

Der Befehl »del« zum Löschen von Dateien – wohl eine der am häufigsten benötigten Anweisungen – ist ebenfalls um eine sehr sinnvolle Option erweitert worden. Der Schalter »/p« bewirkt, daß Sie das Löschen jeder ausgewählten Datei mit der Taste [J] bestätigen müssen. Bei einem pauschalen

```
del *.txt /p
```

laufen Sie also nicht mehr Gefahr, ungewollt eine wichtige Textdatei zu verlieren.

dosshell

Syntax: DOSSHELL

Dieser neue Befehl ist in Form einer Stapeldatei vorhanden, die während der Installation des Betriebssystems mit »select menu« (siehe unten) automatisch erzeugt wird. Der Aufruf von »dosshell« bewirkt einen Start der grafischen Benutzeroberfläche von MS-DOS 4.01.

fastopen

Syntax: FASTOPEN <Laufwerk> [=(<n>,<m>)] [...] [/X]

Die Anweisung »fastopen« erhöht die Geschwindigkeit von Disketten- und Festplattenzugiffen, indem sie eine Liste der Positionen der zuletzt benutzten Dateien anlegt. Hierbei gibt »n« die maximale Zahl der Listeneinträge für das Laufwerk an, und »m« steht für die Zahl der Speicherpuffer, die zum Speichern aktueller Dateidaten der zuletzt verwendeten Dateien reserviert werden sollen (also eine Art Cache-Speicher). Mit der neuen Option »/x« werden diese Speicherpuffer im Expanded Memory (nach LIM-Standard 4.0) angelegt, falls ein solcher erweiterter Speicher zur Verfügung steht.
Wenn Sie »fastopen« über den Install-Befehl in der Datei »config.sys« aufrufen möchten, dann muß der entsprechende Gerätetreiber für den Erweiterungsspeicher schon vorher geladen sein.

graphics

Syntax: GRAPHICS [<Drucker>] [<Profil>] [/R] [/B] [/LCD] [/PB:<x>]

Der Graphics-Befehl erlaubt den Ausdruck des Grafikbildschirms durch Drücken der Taste [PrtScr] beziehungsweise [Druck]. Der Platzhalter »Profil« steht hierbei für die Datei, in der die Daten aller angeschlossenen Drucker enthalten sind (normalerweise »graphics.pro«). In der Version 4.01 des Betriebssytems kamen außer der Unterstützung des neuen Grafikadapters »8514/A« von IBM einige weitere Neuerungen hinzu:
»graphics.pro« enthält nun auch die sogenannte »Printbox-Information« (abgekürzt »PB«), die die Größe des Druckbereichs (»Printbox«) bestimmt. Hierzu sind in der Datei »graphics.-pro« entsprechende Daten zu allen

IBM-Druckern gespeichert, die Sie mit der Befehlsoption »<Drucker>« (»color1«, »color4«, »color8«, »graphics«, »graphicswide« und »thermal«) auswählen können. Die Option »/pb:x« wählt die entsprechende »Printbox« aus, wobei »x« für »std« (»Standard«) oder »lcd« (wie die Option »/lcd«) steht.
Je nach verwendetem Bildschirmtyp und angeschlossenem Drucker wird der Ausdruck der Grafik optimal gestaltet und bei Platzproblemen auf der Druckseite eventuell um 90 Grad gedreht.

mem

```
Syntax: MEM [/PROGRAM] [/DEBUG]
```

Mit diesem neuen Befehl steht unter MS-DOS ab Version 4.01 endlich ein Hilfsprogramm zur Verfügung, das außer der Größe des noch verfügbaren Arbeitsspeichers auch weitreichendere Informationen über die augenblickliche Belegung des Speichers anzeigt. Rufen Sie »mem« ohne weitere Optionen auf, so erhalten Sie eine genaue Angabe über den in Ihrem Computer installierten Arbeitsspeicher, wobei Extended wie auch Expanded Memory berücksichtigt werden. Verwenden Sie zusätzlich eine der beiden oben genannten Optionen, so erhalten Sie außerdem noch genaue Informationen über die momentane Speicherbelegung, also über speicherresidente Programme und installierte Gerätetreiber (die ausführlichste Information erhalten Sie mit »/debug«).

mode

```
Syntax:
MODE [<Gerät>] [<diverse Parameter>]
[RETRY=<c>]
MODE CON RATE=<r> DELAY=<d>
```

Der Befehl »mode« erlaubt das Einstellen diverser Parameter für serielle und parallele Schnittstellen, Bildschirm, Tastatur, Codepages und vieles mehr, so daß auf eine genaue Angabe der sehr umfangreichen Syntax hier verzichtet wird. Eine interessante Neuerung bei MS-DOS 4.01 ist die Option »/retry = c«, die bei den meisten Anwendungen von »mode« mit angegeben werden kann. Für den Parameter »c« setzen Sie entweder »e« (»error«), »b« (»busy«), »r« (»ready«) oder »n« (»none«) ein. »retry« steuert entsprechend das Verhalten des Computers bei Zeitfehlern während des Zugriffs auf eine Schnittstelle. Geben Sie die Option »retry« bei »mode« mit an, so versucht der Rechner bei Auftreten eines Fehlers ständig weitere Zugriffe auf die entsprechende Schnittstelle.
Wählen Sie für »c« die Option »e«, so zeigt der Computer einen bei einer Statusüberprüfung der Schnittstelle aufgetretenen Fehler an. Um einen Belegtstatus (»busy«) auszugeben, dient der Schalter »b«, wohingegen »r« die Arbeitsbereitschaft der angesprochenen Schnittstelle anzeigt. Mit »n« weisen Sie den Computer an, keinen erneuten Zugriff auf eine fehlerhafte Schnittstelle durchzuführen.
Eine weitere Neuerung betrifft die Steuerung der Tastatur durch Angabe der Parameter »rate« und »delay«. Dadurch lassen sich die Häufigkeit der Tastaturabfrage (2 bis 30 Wiederholungen pro Sekunde für Werte von »r« zwischen 1 und 32) und die Verzögerung bis zur Aktivierung der automatischen Wiederholfunktion (erlaubte Werte für »d« sind »1«, »2«, »3« und »4«, was 1/4, 1/2, 3/4 und 1 Sekunde Verzögerung entspricht) einstellen.

replace

```
Syntax: REPLACE [<Quellverzeichnis>]
[<Zielverzeichnis>] [/A] [/P] [/R]
[/S] [/W] [/U]
```

Die Anweisung »replace« ersetzt die Dateien im Zielverzeichnis durch Dateien aus dem Quellverzeichnis, die den gleichen Namen haben. Dieser Befehl mit den vielen Optionen existiert auch in älteren Versionen von MS-DOS.
Neu hinzugekommen ist die Option »/u«. Dieser Schalter veranlaßt »replace«, nur solche Dateien aus dem Zielverzeichnis zu ersetzen, die älter sind als die gleichnamigen Dateien im Quellverzeichnis. Hierzu werden Datums- und Uhrzeiteintrag aus dem Inhaltsverzeichnis verglichen.

select menu

```
Syntax: SELECT MENU
```

Der Befehl »select« kann unter MS-DOS 4.01 nur als menügeführtes Installationsprogramm aufgerufen werden. In früheren Versionen von MS-DOS diente er lediglich dazu, das Betriebssystem zusammen mit den Daten für eine landesspezifische Installation auf eine Diskette zu übertragen. In der Version 4.01 von MS-DOS wurde »select« erheblich erweitert und dient primär der Komplettinstallation des Betriebssystems von den Originaldisketten aus.
Wenn Sie die Installationsdiskette von MS-DOS 4.01 in das Diskettenlaufwerk legen und den Rechner neu booten, wird »select« automatisch gestartet. Mit »select menu« können Sie das Installationsprogramm auch von der MS-DOS-Kommandozeile aus aufrufen. Das Select-Programm arbeitet durchweg interaktiv und erlaubt dem Anwender, das Betriebssystem gemäß seinen Wünschen an den Computer anzupassen. Zuletzt erzeugt »select« auch die Dateien »autoexec.bat«, »config.sys« und »dosshell.bat«, die das Betriebssystem beim Booten in den gewünschten Ausgangszustand bringen, beziehungsweise die grafische Benutzeroberfläche »Dosshell« starten.
Allerdings befinden sich nach der Installation zwei »Schönheitsfehler« in den Dateien »autoexec.bat« und »config.sys«. Ersetzen Sie in »config.sys« in der Zeile, die den Tastaturtreiber »keyb.com« aufruft, die Zeichenfolge

Toolboxen für Programmierer gesucht

Haben Sie eine Toolbox in C, Pascal, Basic, dBase, Clipper oder Assembler geschrieben? Beispielsweise schnelle, universelle Grafikroutinen, eine Textverarbeitungs- oder eine Datenbank-Toolbox? Dann sollten Sie sich schnellstmöglich mit uns in Verbindung setzen. Schicken Sie uns unverbindlich eine Diskette mit Ihrem Programm und einen Ausdruck der Dokumentation.
Falls Sie sich vorweg informieren möchten: Zwischen 15.00 und 17.00 Uhr stehen wir Ihnen an allen Werktagen unter der Rufnummer (089) 4 61 00 10 zur Verfügung.

Schicken Sie Ihre Unterlagen an folgende Adresse:
Redaktionsbüro Everts & Hagedorn
»DOS Software«
Hans-Stießberger-Str. 3
8013 Haar/München

»US« durch »GR«. Der Tastaturtreiber wird dann nochmals in »autoexec.bat« installiert. Daher können Sie diese Zeile ersatzlos streichen.

tree

Syntax: TREE [<Laufwerk>] [/F] [/A]

Diese Anweisung ist schon seit langem Bestandteil von MS-DOS. Leider war die optische Gestaltung des Unterverzeichnissystems in früheren MS-DOS-Versionen äußerst mangelhaft. Der neue Tree-Befehl ist nun wirklich ein »Baum«: Die Verzeichnisstruktur des gewählten Laufwerks wird mit Hilfe der IBM-Grafikzeichen übersichtlich in Form eines »Verzeichnisbaums« angezeigt. Die Befehlsoption »/a« bewirkt, daß zur Darstellung des Verzeichnisbaums nur Zeichen verwendet werden, die auf allen Codeseiten zur Verfügung stehen, wodurch die Bildschirmausgabe beschleunigt wird.
Wie auch schon bei früheren Versionen, zeigt der Tree-Befehl auf Wunsch nicht nur die Verzeichnisstruktur, sondern auch die darin enthaltenen Dateien an. Dazu dient der Parameter »/f«.

vol

Syntax: VOL [<Laufwerk>]

»vol« zeigt den Namen und die Datenträgernummer des gewählten Laufwerks an.
Ab der Version 4.01 vergibt MS-DOS beim Formatieren automatisch eine »Datenträgernummer«, die zur Identifizierung eines Massenspeichers dient. Es kann weiterhin zusätzlich der schon bekannte »Datenträgername« (»label«) angegeben werden. Diesen können Sie entweder beim Formatieren durch die Option »/v:<Name>« festlegen oder nach Beendigung eines Formatiervorgangs eingeben. Ab MS-DOS 3.3 besteht außerdem durch die Anweisung »label« die Möglichkeit, den Datenträgernamen nachträglich zu verändern.
Zur Kontrolle Ihrer Betriebssystemdisketten sind in der abgedruckten Tabelle sämtliche Dateien aufgelistet, die sich auf den MS-DOS- beziehungsweise PC-DOS-Disketten befinden müssen. Bitte beachten Sie, daß viele Hersteller zusätzlich Textdateien mit weiteren Informationen auf den Disketten speichern, die Sie unbedingt lesen sollten.
Oftmals finden Sie auf den Disketten auch noch speziell an die Hardware Ihres Computers angepaßte Steuerprogramme für die Speicherverwaltung (»EMS-Treiber«) oder andere Gerätetreiber mit der Namenserweiterung »sys«. Im Zweifelsfall sollten Sie das technische Handbuch zu Ihrem Computer zu Rate ziehen, um diese Programme korrekt installieren zu können. (Thomas Rueß/tr)

Konfigurationsdateien:	
autoexec.bat	Stapeldatei, die beim Systemstart ausgeführt wird
config.sys	Konfigurationsdatei, die beim Systemstart gelesen wird
Gerätetreiber:	
ansi.sys	Gerätetreiber für die genormte Bildschirmausgabe (Ansi-Norm)
country.sys	Treiber mit länderspezifischen Konfigurationsdaten; wird für den Country-Befehl der Datei »config.sys« benötigt
display.sys	Gerätetreiber für die Bildschirmausgabe
driver.sys	Gerätetreiber für externe Massenspeicher
4201.cpi	Steuerdateien für den Bildschirmtreiber »driver.sys«
4208.cpi	
5202.cpi	
ega.cpi	
lcd.cpi	
himen.sys	Dieser Gerätetreiber stellt ein 64 KByte großes Speichersegment ab Adresse A000:0000hex für MS-DOS zur Verfügung. Allerdings unterstützt nur Microsoft Windows diesen zusätzlichen Speicher. Der Treiber ist nicht in allen Versionen von MS-DOS 4.01 enthalten.
printer.sys	Gerätetreiber für die Druckerausgabe
ramdrive.sys	Treiber zur Installation einer RAM-Disk; bei PC-DOS von IBM heißt dieser Treiber »vdisk.sys«.
smartdrv.sys	Treiber für Cache-Speicher (wird nur bei MS-DOS 4.01 mitgeliefert)
vdisk.sys	Treiber zur Installation einer RAM-Disk unter PC-DOS 4.01 (siehe »ramdrive.sys«)
xma2ems.sys	Treiber für EMS-Speicher
xmaem.sys	Treiber zur Verwaltung von Extended Memory
MS-DOS-Hilfsprogramme (Com-Dateien):	
assign.com	Umleitung von Laufwerkszugriffen
backup.com	Anlegen von Sicherheitskopien einer Festplatte
basic.com	Basic-Interpreter von PC-DOS
basica.com	Basica-Interpreter von PC-DOS
mortgage.bas	Demonstrationsprogramm für den Basica-Interpreter von PC-DOS
chkdsk.com	Überprüfen der Dateistruktur
command.com	MS-DOS-Kommandoprozessor
comp.com	Vergleichen von Dateien
debug.com	Fehlersuche in Assemblerprogrammen
diskcomp.com	Vergleichen von Disketten
diskcopy.com	Kopieren von Disketten
edlin.com	MS-DOS-Zeileneditor
fdisk.com	Einrichten von Festplattenlaufwerken
format.com	Formatieren von Massenspeichern
graftabl.com	Darstellung des kompletten Zeichensatzes auf dem Bildschirm
graphics.com	Ausgabe eines Grafikbildschirms auf den Drucker
graphics.com	Grafikprofile mit Daten zu verschiedenen Druckermodellen für »graphics.com«
keyb.com	Installation eines landesspezifischen Tastaturtreibers
keyboard.sys	Übersetzungstabellen für »keyb.com«
label.com	Ändern eines Datenträgernamens
mode.com	Einstellung verschiedener Systemparameter
more.com	Seitenweises Ausgeben von Informationen auf dem Bildschirm
print.com	Druckerausgabe von Dateien
recover.com	Wiederherstellung defekter Dateien (mit Vorsicht zu genießen!)
restore.com	Zurückschreiben von mit »backup« angefertigten Sicherheitskopien
select.com	Installation des Betriebssystems
sys.com	Kopieren der Betriebssystemdateien
tree.com	Darstellung der Verzeichnisstruktur
MS-DOS-Hilfsprogramme (Exe-Dateien):	
append.exe	Festlegung eines Suchpfades für Datendateien
attrib.exe	Ändern von Dateiattributen
exe2bin.exe	Übertragen von Programmdateien ins Binärformat
fastopen.exe	Beschleunigen von Massenspeicherzugriffen
fc.exe	Vergleichen von Dateien
find.exe	Auffinden von Zeichenketten
gwbasic.exe	GW-Basic-Interpreter von MS-DOS
join.exe	Umleiten von Laufwerkszugriffen
mem.exe	Anzeige der momentanen Arbeitsspeicherbelegung
nlsfunc.exe	Verwalten von Codeseiten
replace.exe	Ersetzen älterer Dateien
select.exe	MS-DOS-Hilfsprogramm zur Installation des Betriebssystems
select.dat	Datendatei für »select«
select.hlp	Datendatei für »select« (Hilfsinformationen)
select.prt	Datendatei für »select«
share.exe	Dateiverwaltung in Netzwerken
sort.exe	Filterprogramm für die sortierte Ausgabe von Daten
subst.exe	Umleitung von Laufwerkszugriffen
xcopy.exe	Kopieren kompletter Unterverzeichnisse
Dateien der »Dosshell«:	
shellb.com	
shellc.exe	
shell.clr	
shell.hlp	
shell.meu	
dosutil.meu	
pcibmdrv.mos	Maustreiber für die »Dosshell« (nur bei PC-DOS)
pcmsdrv.mos	Maustreiber für die »Dosshell« (nur bei PC-DOS)
pcmspdrv.mos	Maustreiber für die »Dosshell« (nur bei PC-DOS)

Tabelle. Diese Dateien befinden sich auf den Systemdisketten Ihres MS-DOS-4.01-Betriebssystems

Komfort ab Werk

Die Dosshell von MS-DOS 4.01

Zur Version 4.01 des MS-DOS-Betriebssystems liefert Microsoft eine grafische Benutzeroberfläche mit. Nur die wenigsten Anwender beschäftigen sich damit, denn meist hat man sich bereits an ein anderes Programm gewöhnt. Dennoch ist es lohnenswert, sich zumindest mit den Fähigkeiten der Dosshell vertraut zu machen.

Wenn Sie die Installation von MS-DOS 4.01 automatisch durchführen lassen (dies geschieht durch den Befehl »select menu«) und alle Fragen nach Ihrer Systemkonfiguration beantwortet haben, wird MS-DOS die wichtigsten Programme von den Betriebssystemdisketten auf die Festplatte kopieren. Unter anderem legt es dabei eine Datei an, die Sie auf den Disketten selbst nicht finden werden, nämlich »dosshell.bat«. Hierbei handelt es sich um eine Stapeldatei, welche die neue grafische Benutzeroberfläche »Dosshell« startet. Die Stapeldatei hat in etwa folgendes Aussehen:

```
@c:
@cd c:\dos
@shellb dosshell
@if errorlevel 255 goto end
:common
@break=off
@shellc /mos:pcibmdrv.mos /tran /color
/dos /menu /mul /snd /meu:shell.meu
/clr:shell.clr /prompt /maint /exit
/swap /date
:end
@break=on
```

Geringe Änderungen können sich je nach den bei der Installation von MS-DOS angegebenen Parametern ergeben.

Die verschiedenen Parameter hinter der Shellc-Anweisung bestimmen unter anderem, welche Funktionen innerhalb der Dosshell aktiviert sein sollen. Es ist nämlich möglich, beispielsweise jeglichen Zugriff von der Dosshell aus auf die MS-DOS-Betriebssystemebene zu verhindern.

Hierzu lassen Sie den Parameter »/exit« einfach weg, und schon können Sie die Dosshell nicht mehr verlassen. Solche speziellen Einstellungen sind eventuell nützlich, wenn viele Personen am gleichen Computer arbeiten und es verhindert werden soll, daß Unbefugte fremde Daten lesen können. Normalerweise ist dies jedoch nicht nötig.

Die Dosshell bietet noch weitere Funktionen, um Zugriffsrechte an verschiedene Benutzer zu verteilen. Dazu später mehr.

Sind Sie Besitzer einer Maus, die nicht von den Dosshell-Maustreibern mit der Endung »mos« unterstützt wird, so müssen Sie in »dosshell.bat« eine kleine Änderung vornehmen: Entfernen Sie »/mos:pcibmdrv.mos« aus der Zeile mit der Shellc-Anweisung (hier kann auch »pcmsdrv.mos« oder »pcmspdrv.mos« stehen), und setzen Sie dafür an den Anfang der Stapeldatei den Aufruf Ihres eigenen Maustreibers, der auf den Disketten im Lieferumfang Ihrer Maus zu finden ist. Meist trägt er den Namen »mouse.com«.

Haben Sie so alle Hürden der Installation bezwungen, starten Sie die neue Dosshell durch Eingabe von »dosshell«. Nach kurzer Zeit sollte ein Eingangsmenü ähnlich wie in Bild 1 auf Ihrem Bildschirm erscheinen.

Einige der Elemente auf dem Bildschirm werden immer sichtbar sein. Dies ist vor allem der Doppelbalken (Menübalken) in den obersten beiden Bildschirmzeilen, der Datum und Uhrzeit anzeigt und als »Menü-Identifikator« dient. In der Mitte des obersten Balkens ist stets der Name des gerade aktiven Menüs zu lesen, in diesem Fall »Programme starten«. Dieses Menü ist das sogenannte Startmenü.

In der dritten Zeile finden Sie die Bezeichnung »Hauptgruppe« und in den folgenden Zeilen die Menüpunkte »Systemanfrage«, »Dateisystem«, »Farben ändern« und »DOS-Dienstprogramme...«.

Um die Dosshell zu verstehen, muß man sich zunächst etwas mit ihrer »Philosophie« vertraut machen: Unter MS-DOS gibt es Unterverzeichnisse, die ihrerseits Unterverzeichnisse oder auch Dateien enthalten können. Dies dient vor allem dazu, den Überblick über große Datenmengen nicht zu verlieren. Dieses vernünftige Konzept wurde praktisch ohne Änderungen auf die Menüstruktur der Dosshell übertragen, nur daß an die Stelle der Unterverzeichnisse sogenannte »Gruppen« treten.

Auch unter MS-DOS 4.01 wird eine Festplatte weiterhin in Unterverzeichnisse unterteilt, die zunächst einmal nicht das geringste mit den Gruppen der Dosshell zu tun haben.

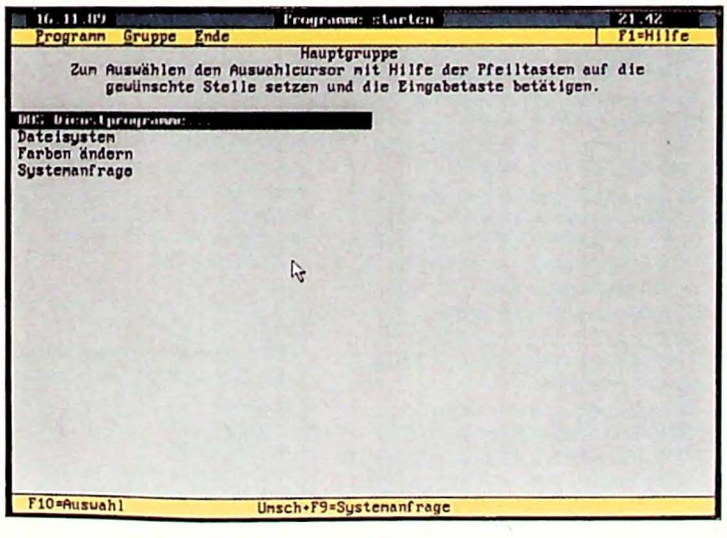


Bild 1. Das Startmenü der Dosshell präsentiert sich zunächst noch fast leer. Es enthält eine Untergruppe und drei ausführbare Programme.

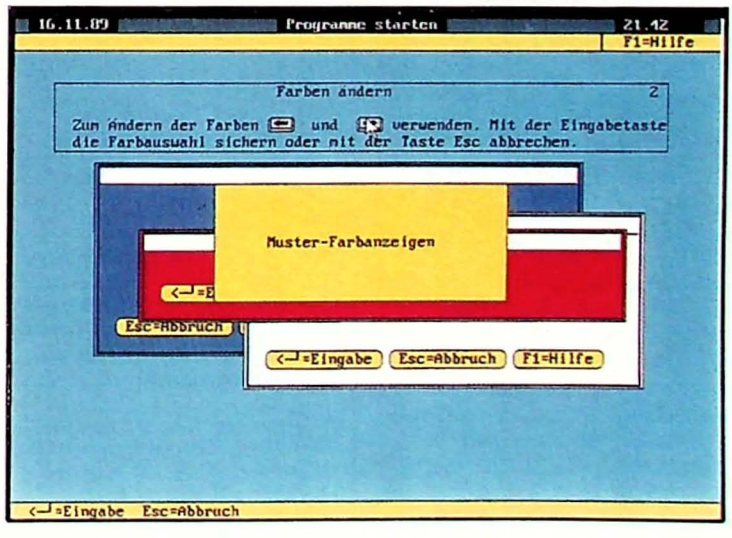


Bild 2. Das Programm »Farben ändern« erlaubt eine individuelle Anpassung des Dosshell-Erscheinungsbildes

Der direkte Weg zu besseren Programmen

Erst prüfen, dann kaufen:
Schauen Sie sich jedes Werk in Ruhe an: 10 Tage lang dürfen Sie Ihr Ansichtsexemplar unverbindlich zu Hause prüfen.

Besuchen Sie uns auf der
CeBIT'90
21. – 28. MÄRZ 1990
Halle 8 · Stand C48 und
Halle 22 · Stand B41

Lösungen in Turbo-Pascal 4.0, 5.0 und 5.5

Dieses neue Nachschlagewerk bietet Ihnen:

- **detailliertes Know-how zum Turbo-Pascal-System;**
- **den Lehrgang Software-Engineering;**
- **Betriebssystemroutinen,** u. a. ROM-BIOS und IBM-Hardware (z. B. Druckertreiber für Turbo-Pascal V 4.0 bis 5.5, asynchrone Datenübertragung, Routinen zur Druckeransteuerung), DOS-Zugriff;
- **hardwarenahe Programmierung:** u. a. Rechnerkopplung, Floppytuning, Drucker-/Bildschirmwechsel mit Turbo V4.0 bis 5.5;
- **raffinierte Utilities: Schnittstellenutilities** (u. a. Turbo-Inliner, Lesen von dBase III-Dateien), **Grafikutilities** (digitale Bildverarbeitung, Grafiken ausdrucken), **Dateiverwaltungs-Utilities** (Ver- und Entschlüsseln von Dateien);
- **grafische Benutzeroberfläche SISGRAF:** GEM-ähnliche Benutzeroberfläche zur Einbindung in eigene Programme oder zur Programmierung von Systemsteuerung wie Windows;
- **Grafik-Editor** zum Zeichnen und Ausdrucken von Linien oder Rechtecken und zur Flächenbearbeitung, **Zeichensatz- und Text-Editor;**
- **Anwenderprogramme für Technik/Mathematik,** u. a. lineare Algebra, statistische Qualitätskontrolle;
- **betriebswirtschaftliche Anwendungen:** u. a. Finanzbuchhaltung, Restschuldberechnung;
- **Turbo-Pascal-Modulbibliotheken,** u. a. String- und DOS-Routinen (z. B. Dialogverarbeitung im Grafik-Modus, Desktop-Steuerung), Mausfunktionen und Interrupts, Terminate Stay Resident Programmierung (TSR-P);
- **Erweiterungsausgaben zum Grundwerk** mit neuen Utilities, Tools und Musterprogrammen für Turbo Pascal 5x.

Fordern Sie am besten gleich mit nebenstehender Karte an:

Musterprogramme, Utilities und Modulbibliotheken für raffinierte Lösungen in Turbo-Pascal und Weiterentwicklungen

stabiler Ringbuchordner, Format DIN A4, Grundwerk ca. 780 Seiten, inkl. 3 Disketten, Best.-Nr.: 2500/72500, zum Preis von DM 98,–.

Alle 2 – 3 Monate erhalten Sie Erweiterungsausgaben zum Grundwerk mit jeweils ca. 130 Seiten zum Seitenpreis von 39 Pfennig (Abbestellung jederzeit möglich).

Mehr Erfolg mit DOS durch effizienten Einsatz der Systemroutinen

Dieses neue Nachschlagewerk bietet Ihnen:

- **sämtliche DOS-Funktionen für gezieltes und sicheres Programmieren;**
- **Programme und Utilities zum Soforteinsatz auf Diskette;**
- **Interruptprogrammierung:** detaillierte Funktionsbeschreibungen zu BIOS-Interrupts und DOS-Funktionen und deren gezielte Anwendungen;
- **Das DOS-System im Detail:** genaue Beschreibungen der DOS-Versionen einschließlich DOS 4.0; ausführliche Erläuterungen zu den Funktionen der Systemkomponenten;
- **DOS-Kommandos:** Alphabetisch und funktionsorientierte Befehlsverzeichnisse mit ausführlichen Erläuterungen und Beispielen;
- **Multitasking unter DOS:** Schrittweise Entwicklung einer Betriebssystem-Erweiterung, die als Zusatz von DOS installiert wird und Echtzeit-Multitasking auf dem PC realisiert. Die Programmierumgebung von DOS bleibt voll kompatibel;
- **Musterlösungen zum Soforteinsatz,** u. a. komplette Systemprogramme zur Ansteuerung der PC-Peripherie, Parameterübergabe und Datentransfer zwischen Programmen, Bausteinprogrammierung, Dateiverwaltung, Steuerung interner Abläufe;
- **DOS-Utilities und Tools,** z. B. Bildschirm-Hardcopy direkt aus Turbo-Pascal-Programmen aufrufen, Installierung von ANSI-Escape-Sequenzen in Batch-Dateien und komplette Tools als leistungsstarke Alternativen zu den DOS-Werkzeugen.

Fordern Sie noch heute an:

DOS-Systemprogrammierung für IBM-PCs und Kompatible

stabiler Ringbuchordner, Format DIN A4, ca. 400 Seiten, inkl. Diskette, Bestell-Nr. 5000 75000, Preis: DM 98,–.

Alle 2 – 3 Monate erhalten Sie Erweiterungsausgaben zum Grundwerk mit jeweils ca. 130 Seiten zum Seitenpreis von 39 Pfennig (Abbestellung jederzeit möglich).

INTEREST-VERLAG
Fachverlag für Special Interest Publikationen und Anwendersoftware
Industriestraße 21
D-8901 Kissing
Tel. 0 82 33 / 21 01-0

Für Ihre Anforderung verwenden Sie bitte nebenstehende Bestellkarte

Eine Gruppe ist lediglich ein Menü innerhalb der Dosshell, von dem aus andere (untergeordnete) Menüs oder Programme aufgerufen werden können. Die Informationen, welche Programme und Untergruppen zu einer Gruppe gehören, wird von der Dosshell in einer Datei mit der Endung »meu« gespeichert. Innerhalb einer Gruppe können also Programme stehen, die auf die unterschiedlichsten Unterverzeichnisse auf der Festplatte verteilt sind.

Das Startmenü stellt die sogenannte »Hauptgruppe« dar. Diese enthält nach einer Neuinstallation bereits einige Elemente, nämlich drei ausführbare Programme und eine weitere Gruppe. Eine Untergruppe unterscheiden Sie von einem Programmaufruf dadurch, daß nach dem Namen drei Punkte erscheinen.

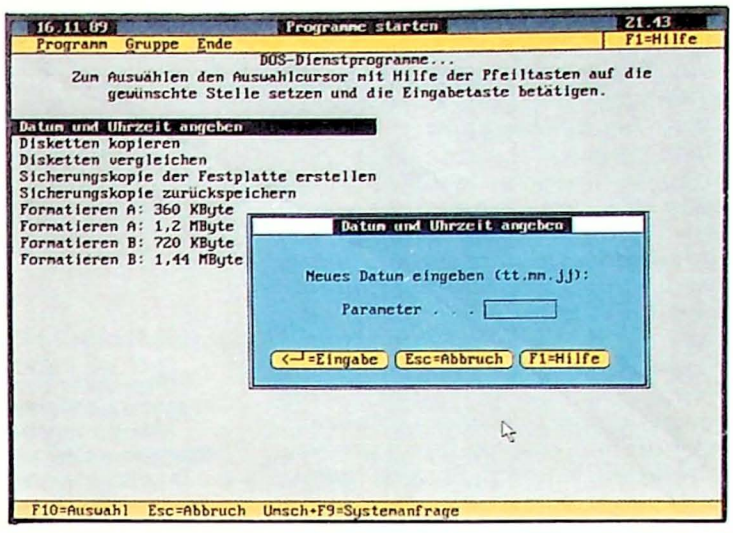


Bild 3. Beim Aufruf von Programmen erscheinen häufig Dialogboxen, über die Daten eingegeben werden können

Zunächst sollten Sie die von der Dosshell verwendeten Farben an Ihre Wünsche anpassen. Dazu müssen Sie das in der Hauptgruppe befindliche Programm »Farben ändern« starten. Setzen Sie den Mauszeiger auf »Farben ändern«, und drücken Sie zweimal hintereinander die linke Maustaste. Sie erhalten daraufhin einen Bildschirm, wie ihn Bild 2 zeigt. Mit den Pfeiltasten »rechts« und »links« können Sie nun einige Farbkombinationen »durchblättern«. Haben Sie eine Farbwahl nach Ihrem Geschmack gefunden, drücken Sie [Enter].

Wie Sie sehen, ist das Ausführen von Programmen in der Dosshell recht einfach. Probieren Sie am besten noch den Punkt »Systemanfrage« aus. Durch diese Anweisung gelangen Sie in die normale MS-DOS-Kommandoebene. Um zur Dosshell zurückzukehren, geben Sie »exit« ein.

Was sich hinter dem Programm »Dateisystem« in der gleichen Gruppe verbirgt, erfahren Sie im weiteren Verlauf dieses Beitrags. Ähnlich einfach wie das Starten von Programmen gestaltet sich die Arbeit mit Gruppen. Setzen Sie den Mauszeiger auf die Gruppe »DOS-Dienstprogramme...«, und betätigen Sie zweimal die linke Maustaste. Sofort erhalten Sie ein neues Menü mit weiteren Programmen. Setzen Sie nun den Mauszeiger zum Beispiel auf »Datum und Uhrzeit angeben«, und drücken Sie wieder zweimal die linke Maustaste. Es erfolgt die Anzeige von Datum und Zeit in einer kleinen Menübox (Bild 3), und Sie haben die Möglichkeit, die Werte neu einzustellen. Wollen Sie dies nicht tun, so wählen Sie das Feld »Abbrechen« und betätigen die linke Maustaste.

Dieses Anwählen eines Menüfeldes durch Setzen des Mauszeigers und Druck auf die linke Maustaste wird auch »Anklicken« genannt. Sie sollten sich dies einprägen, weil es eine allgemein übliche Ausdrucksweise ist, die Sie auch in anderer Fachliteratur finden werden. Auch im weiteren Verlauf dieses Beitrags wird diese Bezeichnung verwendet.

Drücken Sie nun [Esc], um wieder in das Startmenü zu kommen. Um den Umgang mit Gruppen besser zu verstehen, legen Sie am besten selbst einige Gruppen an und füllen diese mit ausführbaren Programmen. Klicken Sie dazu das Feld »Gruppe« im Menübalken an. Es erscheint ein sogenanntes »Pulldown-Menü«, das Ihnen verschiedene Möglichkeiten anbietet. Sie können Gruppen hinzufügen, ändern, löschen oder neu anordnen.

Klicken Sie »Hinzufügen« an, um eine neue Gruppe ins Leben zu rufen. Wieder erscheint eine Menübox, die Sie zur Eingabe verschiedener Daten auffordert. Zunächst sollten Sie sich einen passenden Namen für Ihre Gruppe ausdenken (zum Beispiel »Textverarbeitung«, wenn Sie eine Gruppe mit Texteditoren einrichten möchten) und diesen in das Feld »Ti-

Programmstartbefehl (PSC)	Beschreibung
[ ]	(Leerzeichen) Die aktuelle Befehlszeile erscheint bei jedem Programmaufruf.
[/T"..."]	Ausgabe einer Titelzeile in der Dialogbox beim Programmstart; maximal 40 Zeichen sind erlaubt.
[/I"..."]	Ausgabe einer Hinweiszeile in der Dialogbox beim Programmstart; maximal 40 Zeichen sind gestattet.
[/P"..."]	Gibt die Befehlszeile an; verzichten Sie auf die Angabe dieser Anweisung, so erscheint automatisch die Zeichenkette »Parameter...« in der Befehlszeile.
[/D"..."]	Legt einen Vorgabewert für das Befehlszeileneingabefeld fest.
%n	Gibt einen Wert an, der während des Programmlaufs eingegeben werden muß; eckige Klammern sind bei dieser Anweisung nicht nötig.
[%n]	Speichert einen vorausgegangenen Wert »%n« für die spätere Verwendung; diese Anweisung muß vor allen anderen PSCs in den eckigen Klammern eingegeben werden.
[/D"%n"]	Zeigt einen Vorgabewert im Befehlszeileneingabefeld an; der zuvor mit [%n] gespeicherte Wert wird verwendet.
[/L"x"]	Begrenzt die Länge des Befehlszeileneingabefeldes auf x Zeichen; der maximale Wert für »x« beträgt 127.
[/R]	Löscht den Vorgabewert im Befehlszeileneingabefeld, wenn die erste vom Anwender gedrückte Taste keine Editiertaste ist.
[/M"e"]	Diese Anweisung überprüft, ob alle angegebenen Programmdateinamen existieren, bevor ein Programmstart ausgeführt wird.
[/C"%n"]	Speichert den Wert »%n« der vorhergehenden Angabe für die Verwendung bei der nächsten Ausführung.
[/F"..."]	Es wird überprüft, ob alle in den Anführungszeichen eingegebenen Dateinamen existieren. Andernfalls wird ein Warnton ausgegeben, und die Dialogbox erscheint erneut.
/#	Übernimmt die Laufwerksbezeichnung vor dem Start des PSCs durch die Dosshell; bei dieser Anweisung dürfen keine eckigen Klammern verwendet werden.
/@	Übernimmt die Pfadbezeichnung aus dem Hauptverzeichnis der Dosshell in den Ablauf des Programmstartbefehls.

In der Tabelle sehen Sie alle Programmstartbefehle, die Sie in der Dosshell verwenden können, um »intelligente« Dialogboxen zu programmieren

tel« eintragen. Die Eingabe schließen Sie wie üblich mit [Enter] ab. In das Feld »Dateiname« tragen Sie einen Namen ein (zum Beispiel »editor«), unter dem diese Gruppe auf der Festplatte gespeichert werden soll. Die Endung »meu« des Dateinamens wird von der Dosshell automatisch vergeben. Wählen Sie ein aussagekräftiges Kürzel, damit Sie die verschiedenen Dateien zur Not auch selbst unterscheiden können. Achten Sie auch darauf, daß kein Dateiname doppelt vorkommt. Halten Sie sich immer vor Augen, daß mit dem Hinzufügen der neuen Gruppe kein neues Unterverzeichnis mit dem von Ihnen gewählten Namen, sondern nur eine Datendatei eingerichtet wird.

Das Feld »Hilfetext« kann einen Text enthalten, der beim Anklicken der Hilfefunktion rechts oben auf dem Bildschirm zusätzliche Informationen für den Benutzer über die Gruppe ausgibt. Sie können dieses Feld jedoch zunächst leer lassen.

Möchten Sie Ihre Daten vor dem Zugriff fremder Personen, die ebenfalls an Ihrem Computer arbeiten, schützen, so fügen Sie in das letzte Feld »Kennwort« ein Paßwort ein. Dieses Paßwort wird immer dann abgefragt, wenn die geschützte Gruppe aktiviert oder verändert werden soll. Ohne dieses Paßwort kann kein Anwender ein Programm aus der Gruppe starten.

Haben Sie alle Eingaben beendet, so drücken Sie [F2], um die Daten zu speichern. In der Hauptgruppe erscheint sofort die eben neu hinzugefügte Gruppe. Klicken Sie die neue Gruppe an, dann erhalten Sie ein (noch leeres) Inhaltsverzeichnis. Hier können Sie nun, genau wie oben beschrieben, weitere Gruppen einfügen.

Wahrscheinlich werden Sie aber zunächst nur einen Programmaufruf einfügen wollen. Wählen Sie hierzu das Feld »Programm« im Menübalken

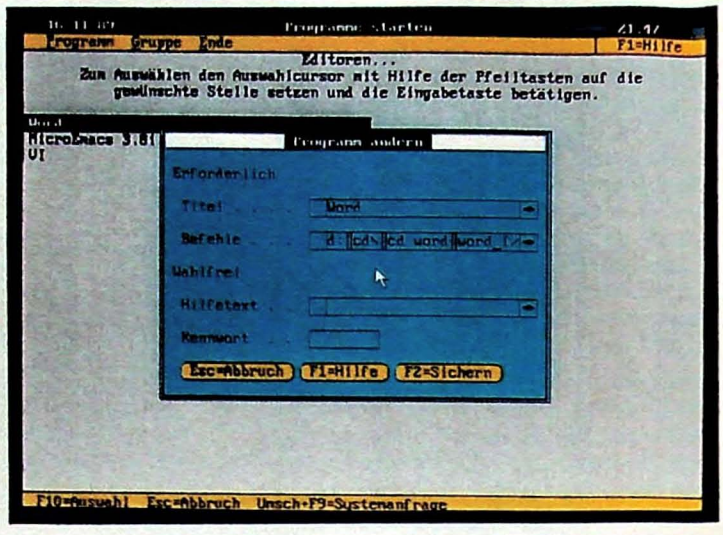


Bild 4. Mit »Programm ändern« können Sie die Startparameter eines Programms editieren

an. Prinzipiell funktioniert das Hinzufügen eines Programms genauso wie das Hinzufügen einer Gruppe, nur daß die Dosshell keine Angabe eines Dateinamens zur Speicherung der Daten fordert. Statt dessen müssen Sie im Feld »Befehle« eine Befehlsfolge eingeben, die das Programm startet. Soll zum Beispiel das Textverarbeitungsprogramm Word aus dem Unterverzeichnis »c:\word« gestartet werden, so geben Sie einfach »c:\word\word« ein.

Speichern Sie anschließend Ihre Eingaben mit [F2]. Sofort erscheint das Programm im Menü der Gruppe. Starten Sie es, indem Sie den Programmnamen zweimal anklicken. Ist alles gut gegangen, wird nun die Dosshell verschwinden und das Programm gestartet.

Solange das Programm aktiv ist, sollten sich keine Unterschiede zum »normalen Betrieb« feststellen lassen. Beenden Sie Ihr Programm ordnungsgemäß, so wird im Anschluß automatisch wieder die Dosshell erscheinen.

Manchmal sind zum Starten eines Programms umfangreichere Befehlsfolgen nötig. Für diesen Fall können Sie beim »Hinzufügen« oder »Ändern« von Programmen im Feld »Befehl« mehrere Befehle hintereinander angeben, die Sie jeweils mit [F4] trennen. Beim Druck auf [F4] erscheint ein senkrechter Doppelstrich (Bild 4).

Es ist außerdem möglich, vor dem Programmstart eine kleine Dialogbox erscheinen zu lassen, die zum Starten des Programms erforderliche Parameter vom Benutzer anfordert. Um dies zu erreichen, müssen in der Zeile »Befehl« entsprechende Anweisungen (sogenannte »program start commands«) eingefügt werden. In Tabelle 1 ist der volle Umfang der verfügbaren Programmstartbefehle aufgelistet.

Auch hier wird die Arbeitsweise am besten durch ein Beispiel klar: Wiederum soll Word gestartet werden, aber diesmal findet zuvor ein Verzeichniswechsel statt, und in einer Dialogbox werden noch einige Eingaben verlangt. Folgende Zeile enthält alle nötigen Befehle:

```
cd\lcd\wordlword [/T"Microsoft WORD 4.0"/I"Word-Parameter angeben, falls erwünscht"]
```

Zuerst wechselt die Befehlsfolge in das Stammverzeichnis »\«, von wo aus anschließend das Unterverzeichnis »\word« aktiviert wird. Die Anweisung »word« startet schließlich das Programm selbst.

Hinter der Anweisung finden nun in eckigen Klammern Programmstartbefehle Platz. Die Anweisung »/T« gibt den in Anführungszeichen folgenden Text (maximal 40 Zeichen) als Menübalken in der beim Programmstart erscheinenden Dialogbox aus. Ein weiterer Text wird mit dem Befehl »/I« innerhalb der Dialogbox ausgegeben (maximal 40 Zeichen) und dient als Erklärung. Die Abfrage von Parametern (»Parameter...«, siehe Bild 5) geschieht automatisch, wenn Sie einen Befehl aus der Tabelle beim Programmaufruf verwenden. In unserem Beispiel könnte als Parameter für Word

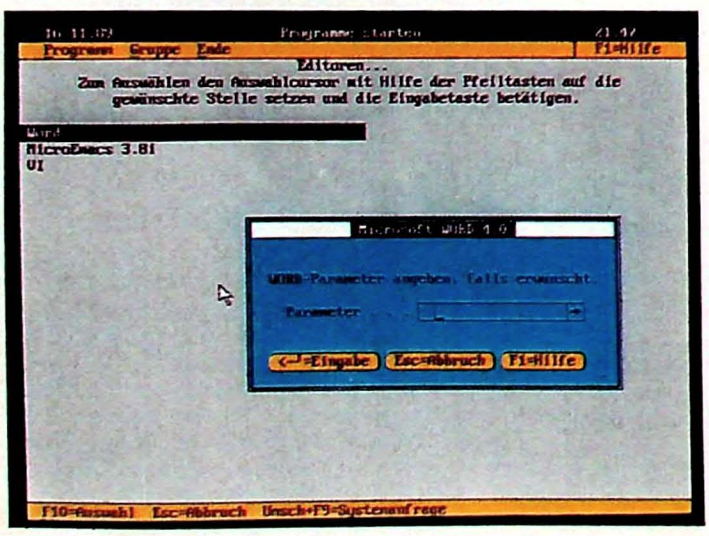


Bild 5. Beim Verwenden von PSCs läßt sich beim Programmstart eine kleine Dialogbox öffnen, die Eingabedaten empfangen kann

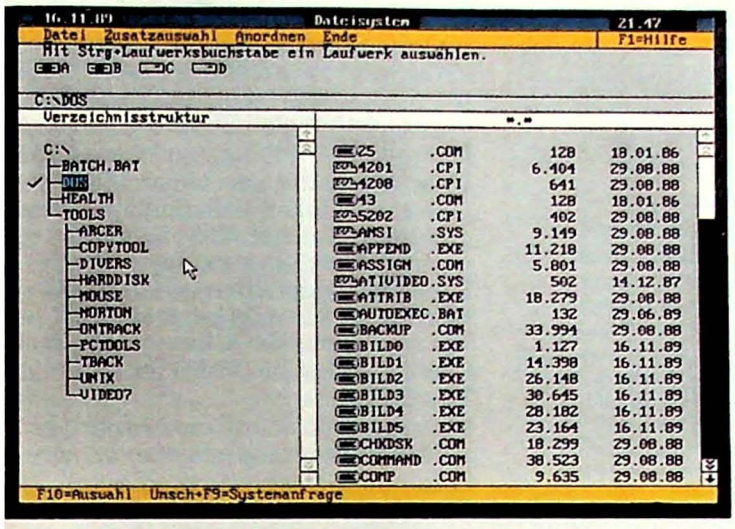


Bild 6. Das Dateisystem der Dosshell stellt die Verzeichnisstruktur und die vorhandenen Dateien nebeneinander dar

»/l« eingegeben werden, wodurch der zuletzt bearbeitete Text wieder geladen wird. Natürlich können Sie jedes andere Programm, das Eingaben von der Kommandozeile liest, auf diese Weise starten.

Installieren Sie nun einige weitere Programme und Gruppen in Ihrer Dosshell. Sie werden sehen, daß es recht einfach ist, ein komfortables Menüsystem mit wenig Arbeit zu implementieren.

Außer einer einfachen Installation Ihres individuellen Programmstartsystems bietet die Dosshell noch ein sogenanntes »Dateisystem«. Wählen Sie hierzu in der Hauptgruppe die Option »Dateisystem« an. Die Dosshell startet daraufhin das Dateisystemunterprogramm, und Sie erhalten einen Bildschirm wie in Bild 6 dargestellt. In der Grundeinstellung sehen Sie auf der linken Seite in einem Bildschirmfenster die Verzeichnisstruktur des aktiven Laufwerks und auf der rechten Seite in einem weiteren Fenster die im gerade aktuellen Verzeichnis enthaltenen Dateien. Im Menübalken erscheinen außerdem Sinnbilder für jedes in Ihren Rechner eingebaute Laufwerk. Um das aktive Laufwerk zu wechseln, klicken Sie einfach das entsprechende Sinnbild einmal an. Um innerhalb des Verzeichnisbaums ein anderes Unterverzeichnis anzuwählen, verfahren Sie wie folgt: Das Anklicken eines Unterverzeichnisses links bringt auf der rechten Bildschirmseite die dort enthaltenen Dateien zur Ausgabe. Sollte Ihr Verzeichnisbaum zu lang sein, um auf den Bildschirm zu passen, können Sie mit Hilfe der rechts an jedem Bildschirmfenster befindlichen »Scroll-Balken« den Fensterinhalt verschieben.

Setzen Sie dazu einfach den Mauszeiger auf das weiße Rechteck, halten Sie die linke Maustaste gedrückt, und bewegen Sie die Maus. Durch Verschieben des weißen Balkens können Sie sich durch den gesamten Verzeichnisbaum arbeiten. Dasselbe gilt für den Fall, daß ein Unterverzeichnis mehr Dateien enthält, als gleichzeitig im rechten Bildschirmfenster Platz finden. Auch hier können Sie mit Hilfe des Scroll-Balkens den Ausschnitt des rechten Fensters verschieben. Die oben und unten am Scroll-Balken befindlichen Felder mit den nach oben und unten deutenden Pfeilen können Sie durch Anklicken dazu verwenden, ein Fenster zeilen- oder seitenweise zu verschieben.

Im rechten Fenster sehen Sie links neben jedem Dateinamen ein Sinnbild, das den Dateityp charakterisiert. Ein ausführbares Programm erkennen Sie an einem ovalen Kreis mit drei Strichen darin (dies soll ein Monitorbild darstellen), eine Datendatei hingegen durch das Symbol »Stück Papier mit Eselsohr«.

Wählen Sie eine beliebige Datei an, indem Sie den Dateinamen einmal anklicken. Die inverse Darstellung des Dateinamens und des Sinnbildes zeigt daraufhin, daß die Dosshell Ihre Wahl akzeptiert hat. Bewegen Sie den Mauszeiger auf den Menübalken in das Feld »Datei«, und klicken Sie es an. Es erscheint wieder ein Pulldown-Menü, das Ihnen verschiedene Möglichkeiten anbietet, um etwas mit der ausgewählten Datei zu unternehmen (Bild 7). Natürlich können Sie, genau wie auf der MS-DOS-Kommandoebene, eine Datei löschen, umbenennen oder kopieren.

Wählen Sie eine dieser Möglichkeiten, so erscheint ein kleiner Dialogkasten, mit dessen Hilfe Sie die entsprechende Funktion ausführen können. So werden Sie zum Beispiel beim Umbenennen von Dateien in der Dialogbox gefragt, wie der neue Dateiname lauten soll.

Manchmal sind in einem Pulldown-Menü einige Optionen in grauer Schrift dargestellt. Diese Befehle sind dann momentan nicht verfügbar. Haben Sie zum Beispiel keine Datei im rechten Fenster ausgewählt, so werden Ihnen die Optionen »Umbenennen«, »Löschen«, »Kopieren« etc. im Dateimenü nicht angeboten, weil die Dosshell schließlich nicht wissen kann, auf welche Datei diese Funktionen angewendet werden sollen.

Können Sie also eine Funktion nicht ausführen, weil sie in grauer Schrift dargestellt ist, so prüfen Sie, ob alle für die Funktion erforderlichen Voreinstellungen getroffen sind. Im unteren Teil des Menüs »Datei« befindet sich eine Anweisung zum Erzeugen von Unterverzeichnissen. Bevor Sie diese Funktion ausführen, müssen Sie im linken Bildschirmfenster das Verzeichnis anwählen, in das Sie ein weiteres Verzeichnis einbauen möchten.

Der Menüpunkt »Gesamtauswahl« erleichtert es Ihnen, alle Dateien in einem Verzeichnis anzuwählen, wenn Sie eine Funktion auf eine große Zahl von Dateien anwenden möchten (zum Beispiel Löschen aller Dateien in einem Unterverzeichnis). Nach Anklicken des Punktes »Gesamtaus-

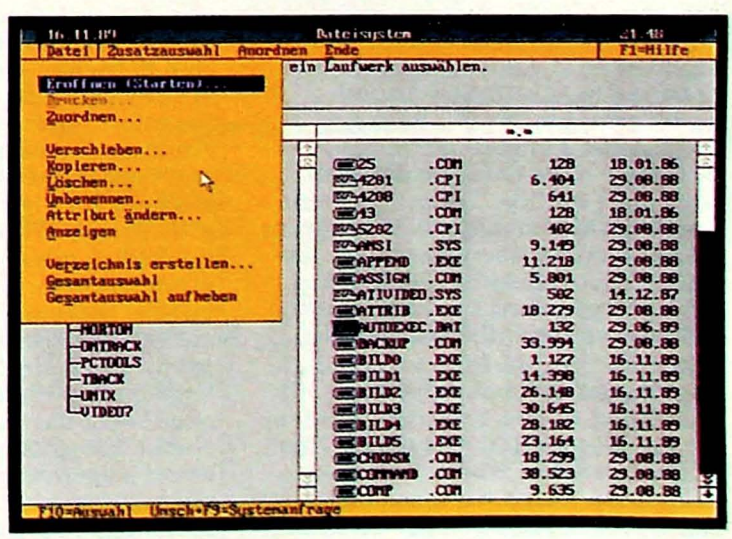


Bild 7. Das Pulldown-Menü »Datei« bietet mehrere Optionen zur Anwendung in beliebigen Dateien

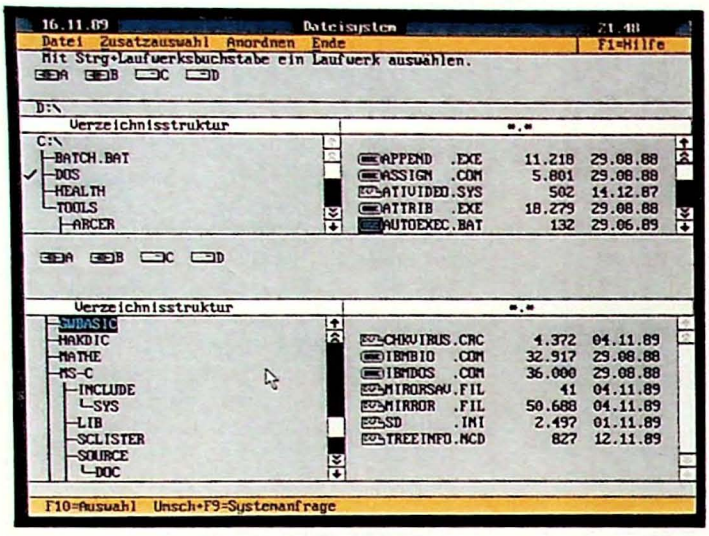


Bild 8. Die gleichzeitige Darstellung von zwei Verzeichnissen geht leider auf Kosten der Übersichtlichkeit, dafür lassen sich zum Beispiel Kopieroperationen sehr bequem ausführen

wahl« erscheinen die Dateikennungssinnbilder aller Dateien invertiert. Mit »Gesamtauswahl aufheben« können Sie alle angewählten Dateien wieder deaktivieren.

Als weitere Möglichkeit steht Ihnen im Menübalken der Programmpunkt »Zusatzauswahl« zur Verfügung. Im dazugehörigen Menü können Sie die Sortierreihenfolge der Dateien für das rechte Bildschirmfenster ändern, Bestätigungen bei Dateioperationen erzwingen und Statusinformationen zu einer Datei ausgeben lassen. Die Ausführung dieser Kommandos erfolgt ebenfalls über weitere Menüs.

Das Untermenü »Anordnen« bietet drei verschiedene Darstellungsmöglichkeiten. Im Modus »ein Verzeichnis« erhalten Sie eine Anzeige wie in der Grundeinstellung – links der Verzeichnisbaum und rechts Dateien aus dem angewählten Unterverzeichnis. Mit »zwei Verzeichnisse« wird der Bildschirm in der Mitte geteilt, und die Dosshell zeigt Ihnen gleichzeitig zwei verschiedene Verzeichnisse, wodurch sich allerdings eine etwas »gedrängte« Bildschirmausgabe ergibt (Bild 8).

Über »ein Laufwerk« heben Sie die Übersichtlichkeit vollständig auf und erhalten auf der rechten Seite die normale Dateiliste, auf der linken Seite jedoch detaillierte Informationen zur gerade angeklickten Datei. Für den normalen Alltagsgebrauch empfiehlt sich die Einstellung »ein Verzeichnis«.

Über das Feld »Ende« wechseln Sie zurück in die Hauptgruppe, um dort Programme auszuführen oder mit Gruppen zu arbeiten. Sie können Programmdateien aber auch direkt aus dem Dateisystem heraus aufrufen, indem Sie den Dateinamen zweimal anklicken. Es erscheint daraufhin eine Dialogbox zur Eingabe von Kommandozeilenparametern.

Wie Sie sicher im Verlauf dieses Beitrags bemerkt haben, besitzt die Dosshell eine ganze Reihe von Fähigkeiten. Sicherlich kann sie sich mit ähnlichen Menüsystemen messen.

(Thomas Rueß/tr)

MS-DOS 4.01 Neue Betriebssystemfunktionen

Seit geraumer Zeit ist das Betriebssystem MS-DOS in der Version 4.01 auf dem Markt. Wer Ambitionen zum Schreiben von System-Software hat, darf sich freuen: Für Programmierer haben »Big Blue« (IBM) und Microsoft interessante neue und erweiterte Funktionen eingebunden.

Wie auch bei älteren MS-DOS-Versionen werden alle Funktionen des neuen Betriebssystems über die Software-Interrupts des PC, AT oder PS/2-Modells angesprochen. MS-DOS verwendet dabei im wesentlichen den Interrupt 21hex. Über eine Funktionsnummer, die dem Interrupt im AH-Register übergeben wird, lassen sich bestimmte Funktionen aus der umfangreichen Bibliothek aller Betriebssystemroutinen auswählen.

Völlig neu bei MS-DOS 4.01 sind die Funktionen 33 (Unterfunktion 5), 63, 69, 6B und 6Chex. Erweitert wurden die Funktionen 33hex, 44hex, 57hex, 5Chex und 65hex. Einige interessante Funktionen wollen wir Ihnen im Rahmen dieses Beitrags genauer vorstellen.

Im Hinblick auf die Kompatibilität zum Betriebssystem OS/2 wurde bei MS-DOS 4.01 die Unterfunktion 5 in Funktion 33hex eingebunden. Diese Funktion diente bislang zum Lesen und Setzen des Unterbrechungskennzeichens für die Tastenkombination [Ctrl-Break]. Wenn Sie nun die Funktion 33hex mit dem Wert 5 im AL-Register aufrufen, so ermitteln Sie damit das Diskettenlaufwerk, von dem aus MS-DOS gestartet wurde. Als Ergebnis des Aufrufs finden Sie im DX-Register den Wert 1 für Laufwerk A, den Wert 2 für Laufwerk B und so weiter. Ein gesetztes Carry-Flag zeigt einen aufgetretenen Fehler an. Ein typischer Aufruf für diese Funktion ist in Listing 1 dargestellt.

Ebenfalls neu ist die Funktion 63hex, die bislang nur in asiatischen Versionen von MS-DOS verfügbar war. Die Funktion wurde für komplexe Sprachen entwickelt, die mehr Zeichen umfassen, als auf der Tastatur darstellbar sind, also beispielsweise für die chinesische oder koreanische Sprache. Ähnlich wie bei der Benutzung der Alt-Taste in Kombination mit dem Ziffernblock zur Nutzung aller ASCII-Zeichen lassen sich mit Funktion 63hex konkrete »Konstruktionsvorschriften« zur Darstellung komplexer Buchstaben geben. Dadurch ist es beispielsweise möglich, Tastencodes für asiatische Schriftzeichen durch die Kombinationen anderer Tasten zu erzeugen. Die Funktion 63hex setzt oder liest einen Vektor (den sogenannten »DBCS-Vektor«) auf eine Umwandlungsroutine für solch komplexe Buchstaben. Diese Routine wird aktiviert, wenn ein spezielles Startzeichen (der sogenannte »Interim Character«) einen komplexen Buchstaben einleitet.

Beim Aufruf der Funktion 63hex müssen Sie im AL-Register angeben, ob der DBCS-Vektor gelesen (AL = 0), das Interim-Flag gesetzt (AL = 1) oder gelesen (AL = 2) werden soll. Wollen Sie das Interim-Flag setzen, müssen Sie zusätzlich den Flag-Code im DL-Register angeben. Abhängig vom gewählten Modus steht nach

dem Aufruf in DS:SI der DBCS-Vektor (für AL = 0) oder in DL das Interim-Flag (für AL = 2). Ein gesetztes Carry-Flag zeigt Ihnen einen Fehler an. Tabelle 1 faßt beide Funktionen noch einmal zusammen.

Neben den neuen Funktionen sind aber auch einige ältere Funktionen von MS-DOS verfeinert und erweitert worden. So gibt es ab MS-DOS 4.01 zwei neue Fehlermeldungen, die sich mit Hilfe der Funktion 59hex abfragen lassen. Zum einen ist dies die Meldung »out of input« und zum anderen »insufficient disk space«. Sie sind durch die erweiterten Fehlercodes 38hex und 39hex gekennzeichnet. Beide Fehlermeldungen sind der Fehlerklasse 4, Fehlerquelle 1, zugeordnet und weisen den Verfahrensvorschlag 1 auf.

Auch die Funktion 5Chex, die für das Schützen von Dateibereichen in Netzwerken zuständig ist, wurde ab MS-DOS 4.01 erweitert. Ab dieser Version ist die Funktion nun in der Lage, auch mehrere Bereiche einer Datei vorübergehend vor Zugriffen zu schützen.

Erweiterungen stellen die einheitliche Verwendung von Klein- und Großschrift sicher

Die Unterstützung nationaler Besonderheiten hat in der Entwicklungsgeschichte des Betriebssystems MS-DOS von Version zu Version einen größeren Stellenwert eingenommen. Auch Version 4.01 macht da keine Ausnahme: Die mit MS-DOS 3.3 eingeführte Funktion 65hex wurde erheblich ausgebaut.

Grundsätzlich dient die Funktion 65hex dazu, landesspezifische Informationen zu verarbeiten. Die Erweiterung umfaßt fünf Unterfunktionen, die der Umwandlung von Klein- in Großbuchstaben dienen. Die Entwickler haben diese Funktionen eingebunden, um eine einheitliche Verwendung von Klein- und Großschrift zu ermöglichen. Mit den neuen Funktionen lassen sich einzelne Zeichen oder ganze Zeichenketten umwandeln. Ferner legen sie fest, welche Umwandlungstabelle Verwendung findet, oder sie ermitteln, ob ein Zeichen überhaupt umgewandelt werden muß.

Bisher ließ sich mit der Unterfunktion 1 der MS-DOS-Funktion 65hex eine komplette Länderinformationstabelle lesen. Neu hinzugekommen sind nun die Unterfunktionen 2 bis 6. Die Unterfunktionsnummern müssen Sie beim Aufruf der Funktion 65hex im AL-Register angeben. Die interessantesten neuen Funktionen sind:

- 2: Ermittle Zeiger auf Tabelle zur Großschrift
- 4: Ermittle Zeiger auf Dateinamentabelle zur Großschrift

Für die korrekte Anwendung der Funktion muß beim Funktionsaufruf das AH-Register den Wert 65hex, das AL-Register den gewünschten Unterfunktionscode, das BX-Register die gewünschte Codeseite und das DX-Register den Ländercode enthalten. Um die aktuelle Codeseite oder den aktuellen Ländercode auszuwählen, verwenden Sie den Wert FFFFhex im entsprechenden Register. Sofern die Codeseite und der Ländercode nicht zusammenpassen, wird ein Fehler (Fehlernummer 2: »file not found«) ausgegeben.

Abhängig von der aufgerufenen Unterfunktion liefert MS-DOS einen Datenblock zurück. Daher muß vor dem Aufruf der Funktion das Registerpaar ES:DI einen Zeiger auf einen Datenpuffer beinhalten, und im CX-Register ist anzugeben, wie viele Bytes dieses Puffers belegt werden dürfen. Die entsprechende Unterfunktion liefert je nach Art der aufgerufenen Funktion einen Informationsblock in diesem Datenpuffer zurück.

Mit den Unterfunktionen 2 und 4 richten Sie die Zeiger auf Tabellen, die zur Umwandlung von Klein- in Großbuchstaben dienen. Dabei sind zwei Tabellen zu unterscheiden: eine zur Umwandlung von »gewöhnlichen« Zeichen in Großbuchstaben und eine zur Umwandlung von Zeichen innerhalb eines Dateinamens in Großbuchstaben. Diese zweite Tabelle wird beispielsweise verwendet, wenn Sie bei der Eingabe eines Dateinamens einen Text in Kleinbuchstaben angeben. Das Betriebssystem sieht dann in der Tabelle nach, welche Kleinbuchstaben in welche Großbuchstaben umzuwandeln sind.

Die Funktionen 2 und 4 ermitteln jeweils einen Zeiger, der auf diese beiden Tabellen verweist. Nach dem Aufruf der Funktion werden 5 Byte in den Datenpuffer an der Adresse ES:DI gespeichert. Das Register CX ist daher vor dem Funktionsaufruf auf den Wert 5 zu setzen. Das erste Byte enthält den Wert 2 oder 4, je nachdem welche Unterfunktion Sie benötigen. Anhand dieses Wertes läßt sich nachträglich feststellen, zu welcher Tabelle der ermittelte Zeiger gehört.

Zwei unabhängige Tabellen gewährleisten eine einfache Zeichekonvertierung

Der Zeiger selbst wird in dem nächsten Doppelwort des Datenpuffers gespeichert. Er enthält die Offset- und Segmentadresse der Tabelle.

Die Tabellen selbst bestehen aus maximal 130 Byte. Das erste Wort der Tabelle gibt dabei die Länge der Tabelle an und darf maximal den Wert 128 aufweisen. Die darauffolgenden Bytes geben die Codes für die bis zu 128 Kleinbuchstaben in Großschrift an. Die Tabelle ist so aufgebaut, daß der letzte Eintrag immer die Konver-

Funktion (hex)	Name	Eingabe (hex)	Ausgabe
33	Query Boot Drive	AH = 33 AL = 5	DX = Laufwerkscode Carry = Fehler
63	Honguel DBCS Support	AH = 63 AL = Modus DL = Interim-Flag	DS:SI = DBCS-Vektor DL = Interim-Flag Carry = Fehler

Tabelle 1. Zwei neue Funktionen von MS-DOS 4.01

Funktion (hex)	Name	Eingabe (hex)	Ausgabe
0	Check Install	AH = B7 AL = 0	AH = 0/1 (nein/ja)
2	Get Version	AH = B7 AL = 2	AX = Versionscode (-1 für MS-DOS)
4	Get Path	AH = B7 AL = 4	ES:DI = Adresse des Pfades
6	Get State	AH = B7 AL = 6	BX = Status (siehe Tabelle 3)
7	Set State	AH = B7 AL = 7	BX = Status (siehe Tabelle 3)
11	Set Return Found Name Mode	AH = B7 AL = 11	

Tabelle 2. Die Schnittstelle zu »append« über die Funktion B7hex des Interrupt 2Fhex

tierung für den ASCII-Wert 255 enthält. Möchten Sie also eine Konvertierung für das ASCII-Zeichen 240 angeben, dann müssen Sie folglich auch Konvertierungen für die ASCII-Werte von 241 bis 255 einrichten. Die Tabelle wäre in diesem Falle 16 Byte lang (zuzüglich 2 Byte für die Länge der Tabelle).

Wenn das umzuwandelnde Zeichen einen Zeichencode aufweist, der kleiner als der Wert »256 – Tabellenlänge« ist, existiert kein Äquivalent in Großschrift. Ist er dagegen größer oder gleich (256 – Tabellenlänge), liegt ein Äquivalent vor. Die Adresse des Zeichencodes läßt sich mit folgender Formel ermitteln:

```
2 + Zeichencode – (256 – Tabellenlänge) +
Startadresse der Tabelle
```

Listing 2 zeigt, wie bei einem gegebenen Zeichencode ermittelt wird, ob ein entsprechendes Äquivalent in Großschrift existiert.

Für Programmierer besonders interessant ist eine Erweiterung von MS-DOS außerhalb des Interrupt 21hex, über den sonst nahezu alle Betriebssystemfunktionen abgewickelt werden: Mit Hilfe der Funktion B7hex des Interrupt 2Fhex sind Sie nunmehr in der Lage, auf den speicherresidenten Teil des Append-Befehls zuzugreifen.

Dieser externe MS-DOS-Befehl dient dazu, neben dem aktuell gewählten Verzeichnis auch noch andere Verzeichnisse nach einer Datei zu durchsuchen. Der Append-Befehl ist dabei, im Gegensatz zum Path-Befehl, nicht auf Dateien mit der Endung »exe«, »com« oder »bat« beschränkt. Je nachdem, in welcher Kombination Sie »append« mit anderen MS-DOS-Anweisungen verwenden, ergeben sich Komplikationen hinsichtlich spezieller (Software-) Schalterstellungen bei einigen Befehlen. Um diese Konflikte zu vermeiden, gibt es ab MS-DOS 4.01 eine dokumentierte Schnittstelle zum residenten Teil von »append«. Sobald »append« einmal aufgerufen wird, installiert sich ein Teil des Programms resident im Speicher. Bei jedem erfolglosen Versuch, eine Datei zu öffnen, untersucht die residente Routine dann automatisch den ihr zugewiesenen Pfad, um die Datei eventuell dort zu finden. Das Anwendungsprogramm, das die Anweisung zum Öffnen der Datei gab, kann allerdings nicht feststellen, ob die Datei im aktuellen Verzeichnis oder im Append-Verzeichnis geöffnet wird.

Die Schnittstelle zu »append« ist mit der Funktion B7hex des Interrupt 2Fhex realisiert. Tabelle 2 und 3 stellen die Unterfunktionen dieser Schnittstelle zusammen.

Bevor Sie auf die Unterfunktionen von »append« zugreifen, ist zunächst zu überprüfen, ob »append« überhaupt aktiv ist, das heißt, ob ein Append-Pfad angegeben wurde. Das geschieht mit der Unterfunktion 0 der Funktion B7hex. Liefert das AL-Register nach deren Aufruf den Wert 0 zurück, so ist »append« nicht installiert.

Die Append-Version läßt sich mit der Unterfunktion 2 ermitteln. Dazu übergeben Sie den Wert FFFFhex (oder -1) im AX-Register und erhalten als Resultat die MS-DOS-Versionsnummer des Append-Befehls. Den von »append« verarbeiteten Pfad erfahren Sie durch die Unterfunktion 4. Einen Zeiger auf diesen Pfad spezifiziert der Interrupt 2Fhex durch das Registerpaar ES:DI. Den Status der Schalterstellungen, die dem Append-Befehl beim Aufruf in der Kommandozeile übergeben werden, ermitteln Sie mit der Unterfunktion 6, während Sie ihn mit der Unterfunktion 7 setzen.

Besonders interessant ist die Unterfunktion 11hex zum Setzen eines Append-internen Schalters. Dieser Schalter zeigt »append« beim nächsten Zugriff an, daß der übergebene Dateiname nicht wirklich verwendet werden soll. Statt dessen dient er dazu, den tatsächlichen Namen der Datei zu ermitteln. Dadurch erfährt das aufrufende Programm bei Verwendung von Dateimasken den konkreten Namen einer Datei. Nach dem Setzen des Schalters mit der Funktion 11hex ist dieser allerdings nur für den nächsten Zugriff aktiv. Er wird danach automatisch zurückgesetzt. Sofern Sie kontinuierlich auf diese Funktion zurückgreifen wollen, müssen Sie den Schalter daher vor jedem Aufruf erneut setzen. Als Anwendungsbeispiel zeigt Listing 3, wie Sie über die Schnittstelle des Interrupt 2Fhex den Append-Pfad auslesen.

Der Statuscode repräsentiert die übergebenen Parameter. Er kann die folgenden Werte annehmen:

BX (hex)	Bedeutung in der Kommandozeile
8000	/X:1
4000	/E
2000	/PATH:ON
0001	ENABLE

Tabelle 3. Der Statuscode der Schnittstelle

Als eine der wichtigsten Änderungen gegenüber den Vorgängern läßt sich bei MS-DOS 4.01 feststellen, daß das Betriebssystem nun auch Festplattenpartitionen mit mehr als 32 MByte anspricht. Diese erweiterte neue Kapazität hat zum Teil erhebliche Auswirkungen auf den Aufruf und die Verwendung von BIOS- und MS-DOS-Interrupts.

Die bisherige Begrenzung von Festplattenpartitionen auf 32 MByte hatte ihre Ursache in den Betriebssystemfunktionen 25 und 26hex. Diese Funktionen werden benutzt, um bestimmte Sektoren der Festplatte (und auch der Diskette) zu lesen. Beim Aufruf von Interrupt 25 oder 26hex ist bis zur MS-DOS-Version 4.01 im Register DX die Nummer des Sektors anzugeben, der von der Festplatte gelesen oder auf die Festplatte geschrieben werden soll. Für die Sektoradresse stehen also 16 Bit (und damit ein Adreßraum von 64 KByte) bereit.

Damit ab MS-DOS 4.01 größere Festplattenpartitionen ansprechbar werden, wurde die Beschränkung der Sektornummer auf 16 Bit aufgehoben. Ab der Version 4.01 des Betriebssystems dürfen Sektornummern für die Interrupts 25 und 26hex 32 Bit belegen. Dadurch taucht aber ein Problem auf: Das Register DX, das bis MS-DOS 4.01 die Nummer des Sektors angab, ist nur 16 Bit breit. Wo bleiben also die zusätzlichen 16 Bit?

Um dieses Problem zu meistern, haben die Entwickler bei IBM beziehungsweise Microsoft einen Trick angewandt, der schon bei der Vorstellung für MS-DOS 2.0 für die Neuordnung der Dateisteuerblöcke (FCBs) verwendet wurde. Wenn beim Aufruf von Interrupt 25 oder 26hex das Register CX den Wert -1 (FFFFhex) enthält, dann hat das zur Folge, daß die Parameter des Interrupts nicht in den Registern, sondern in einem eigenen Parameterblock übergeben werden. Dazu enthält das Registerpaar DS:BX nicht mehr die Adresse des Puffers für den Datenaustausch (dies war bis Version 4.01 der Fall), sondern die Adresse dieses Parameterblocks.

Byte	Bedeutung
0 bis 1	Die unteren 16 Bit der Sektornummer des ersten zu lesenden oder zu schreibenden Sektors
2 bis 3	Die oberen 16 Bit der Sektornummer des ersten zu lesenden oder zu schreibenden Sektors
4 bis 5	Anzahl der zu lesenden Sektoren
6 bis 7	Segmentadresse des Puffers für den Datenaustausch
8 bis 9	Offset-Adresse des Puffers für den Datenaustausch

Tabelle 4. Der Parameterblock für die Interrupts 25 und 26hex

Im Parameterblock geben die ersten vier Byte die Adresse des Sektors, die zwei folgenden Byte die Anzahl der zu lesenden oder zu schreibenden Sektoren und die nächsten vier Byte die Adresse des Puffers für den Datenaustausch an. Tabelle 4 faßt das Format des Parameterblocks zusammen.

Das Besondere an dem neuen Aufruf der Interrupts 25 und 26hex ab MS-DOS 4.01 liegt darin, daß die bisherigen Programme kompatibel bleiben. Der Parameterblock wird nämlich nur bei Festplattenpartitionen benötigt, die wirklich größer als 32 MByte sind. Alle kleineren Partitionen lassen sich über den bisherigen Aufruf der Interrupts 25 und 26hex ansprechen. Die Listings 4 und 5 zeigen Ihnen, wie Sie den Interrupt 25hex in der bisherigen und in der neuen Version anwenden.

Nicht nur die Interrupts 25 und 26hex sind von den Änderungen der Maximalgröße einer Partition betroffen. Alle Interrupts und Funktionen, die Sektornummern übergeben, waren bis zur Version 4.01 auf 16-Bit-Nummern eingeschränkt, benötigen aber nun 32-Bit-Adressen.

Aus dem gleichen Grund, der auch bei den Interrupts 25 und 26hex Anlaß zur Veränderung war, besitzt deshalb auch der Anforderungsblock für Einheitentreiber ab MS-DOS 4.01 ein anderes Format. Dieser Anforderungsblock wird dazu verwendet, von einem Programm aus auf einen Blocktreiber zuzugreifen. Weil Blocktreiber in der Regel Disketten oder Festplattenlaufwerke steuern, waren auch hier aufgrund der größeren Partitionen Änderungen notwendig.

Bis zur Version 4.01 des Betriebssystems gab das letzte Wort in einem solchen Anforderungsblock die Nummer des ersten Sektors an, auf den der Zugriff erfolgen sollte.

Da Partitionen mit einer Größe von mehr als 32 MByte für die Adresse eines Sektors wiederum nicht nur 16 Bit (ein Wort), sondern vielmehr 32 Bit (zwei Worte) benötigen, wurde auch der Anforderungsblock an die größeren Partitionen angepaßt. Ein Wert von -1 (FFFFhex) für die Sektornummer signalisiert, daß 32 Bit zur Adreßbildung verwendet werden sollen. Die erforderlichen zwei Worte werden dann an den »alten« Anforderungsblock angehängt, wodurch dessen Länge um 4 Byte anwächst.

Und schließlich blieben auch die Blocktreiber selbst nicht gänzlich von einer Veränderung ausgenommen. Für ihre Programmierung läßt sich nunmehr auch das bislang unbenutzte Bit 1 des Treiberattributs verwenden. Ist es gesetzt, so zeigt dies an, daß der Treiber in der Lage ist, auch Partitionen größer als 32 MByte zu verwalten.

(Martin Althaus/zi)

```
[HD4]    1: ; Programmname: GET_BD.ASM
[4E3]    2: ; (C)  Everts & Hagedorn
[8G3]    3: ;      DOS International
[P3Ø]    4: ;
[4X6]    5: ;Im Datenbereich muß Platz (1 Word) für
[DB5]    6: ;die Variable BT_DRV sein, die den
[C56]    7: ;Laufwerksbuchstaben aufnehmen soll
[P3Ø]    8: ;
[TK4]    9: get_boot_drive  proc    near
[ØF5]   1Ø:         ;Funktion 33hex, Unterfunktion 5
[UX1]   11:          mov     ah,33h
[AM1]   12:          mov     al,5
[S13]   13:         ;rufe Funktion auf
[3K1]   14:          int     21h
[U15]   15:         ;Fehler, verzweige nach FEHLER
[4W1]   16:          jc      fehler
[P16]   17:         ;addiere Code für ASCII-Darstellung
[GY5]   18:         ;und speichere Laufwerksbuchstaben
[1Ø2]   19:          add     dx,'A'-1
[9P2]   2Ø:          mov     bt_drv,dx
[LZ5]   21:         ;zurück zum aufrufenden Programm
[3PØ]   22:          ret
[BP5]   23:         ;benutze Funktion 59hex für die
[A94]   24:         ;Ermittlung des Fehlers
[R83]   25: fehler: mov     ah,59h
[3T1]   26:          xor  bx,bx
[3K1]   27:          int     21h
[485]   28:         ;zeige dem aufrufenden Programm
[DA2]   29:         ;den Fehler an
[ENØ]   3Ø:          stc
[3PØ]   31:          ret
[RG3]   32: get_boot_drive  endp
```

Listing 1. Die Prozedur »get_bd.asm« stellt das Boot-Laufwerk fest

```
[LF4]    1: ; Programmname: TEST_C.ASM
[4E3]    2: ; (C)  Everts & Hagedorn
[8G3]    3: ;      DOS International
[P3Ø]    4: ;
[4V6]    5: ;In der Speicherstelle Z_CODE steht der
[XC6]    6: ;Zeichencode für ein Zeichen. Es soll
[GX6]    7: ;ermittelt werden, ob es eine Entsprechung
[FK6]    8: ;in der Großschrift-Tabelle besitzt.
[P3Ø]    9: ;
[TZ3]   1Ø: test_char       proc    near
[CN3]   11:         ;Nummer der Funktion
[1Y1]   12:          mov     ah,65h
[S96]   13:         ;Code: Zeiger auf Großschrift lesen
[1Y1]   14:          mov     al,Ø2h
[T53]   15:         ;aktuelle Code-Page
[RF2]   16:          mov     bx,Øffffh
[KT3]   17:         ;aktueller Länder-Code
[VF2]   18:          mov     dx,Øffffh
[V64]   19:         ;5 Byte werden benötigt
[FZ1]   2Ø:          mov     cx,Ø5h
[MF1]   21:          push    ds
[GD1]   22:          pop     es
[T74]   23:         ;Adresse des Datenpuffers
[HR3]   24:          mov     di,offset puffer
[3K1]   25:          int     21h
[D75]   26:         ;Lade Tabellenadresse nach ES:DI
[PB2]   27:          les     di,adrse
[PZ3]   28:         ;lade Länge der Tabelle
[DV2]   29:          mov     bx,es:[di]
[SW3]   3Ø:         ;lade Konstante 256 und
[514]   31:         ;ziehe Tabellenlänge ab
[RX1]   32:          mov     ax,256
[TT1]   33:          sub     ax,bx
[N15]   34:         ;Ist Konvertierung vorhanden?
[KH2]   35:          cmp     ax,z_code
[UR3]   36:         ;nein, springe zu KEINE
[VM1]   37:          jb      keine
[HØ4]   38:         ;lade Zeichencode nach BX
[6L2]   39:          mov     bx,z_code
[UT1]   4Ø:          sub     bx,ax
[CH1]   41:          add     bx,2
[BP2]   42:         ;addiere Offset
[3P1]   43:          add     di,bx
[4V1]   44:          mov     ax,es
[3G5]   45:         ;Übernehme möglichen Übertrag
[4H1]   46:          adc     ax,Ø
[555]   47:         ;nun steht Zeichencode in ES:DI
[3V1]   48:          mov     es,ax
[8V2]   49:         ;lade Zeichencode
[7U2]   5Ø:          mov     al,es:[di]
[ØT1]   51:          xor     ah,ah
[3PØ]   52:          ret
[HH2]   53:         ;zeige Fehler an
[4N3]   54: keine:   mov     ax,Øffffh
[3PØ]   55:          ret
[RX2]   56: test_char       endp
[P3Ø]   57: ;
[W26]   58: ;die folgenden Zeilen gehören in das
[XU5]   59: ;Datensegment, das mit DS adressiert
[DN2]   6Ø: ;werden muß
[P3Ø]   61: ;
[8V2]   62:         ;Code für Zeichen
[F62]   63: z_code  db      ?
[ØX3]   64:         ;Puffer für Funktion
[373]   65: puffer  db      dup (?)
[7Z4]   66:         ;Adresse von Tabellen-Adresse
[512]   67: adrse   dd      ?
```

Listing 2. Die Prozedur »test_c.asm« ermittelt, ob ein Zeichencode in der Übersetzungstabelle enthalten ist

```
[5V4]    1: ; Programmname: LESE_AP.ASM
[4E3]    2: ; (C)  Everts & Hagedorn
[8G3]    3: ;      DOS International
[P3Ø]    4: ;
[MM4]    5: lese_append_pfad   proc near
[ØY4]    6:         ;Funktion für APPEND-Zugriff
[NE4]    7:         ;(Funktion Ø: Install Check)
[262]    8:          mov     ah,Øb7h
[MT1]    9:          xor     al,al
[2M1]   1Ø:          int     2fh
[PL3]   11:         ;Ist es installiert?
[LK1]   12:          cmp     al,Ø
[4X3]   13:         ;nein, springe zu NO_INS
[612]   14:          je      no_ins
[ØY4]   15:         ;Funktion für APPEND-Zugriff
[C54]   16:         ;(Funktion 2: Get Version)
[262]   17:          mov     ah,Øb7h
[7M1]   18:          mov     al,2
[2M1]   19:          int     2fh
[RB4]   2Ø:         ;Ist es ein MS-DOS-APPEND?
[3E2]   21:          cmp     ax,Øffffh
[4X3]   22:         ;nein, springe zu NO_INS
[N62]   23:          jne     no_ins
[ØY4]   24:         ;Funktion für APPEND-Zugriff
[WN4]   25:         ;(Funktion 4: Get Append Path)
[262]   26:          mov     ah,Øb7h
[9M1]   27:          mov     al,4
[2M1]   28:          int     2fh
[W25]   29:         ;zeige fehlerfreie Funktion an
[MT1]   3Ø:          xor     al,al
[3PØ]   31:          ret
[HH2]   32:         ;zeige Fehler an
[9Ø3]   33: no_ins   mov     al,1
[3PØ]   34:          ret
[MZ3]   35: lese_append_path   endp
```

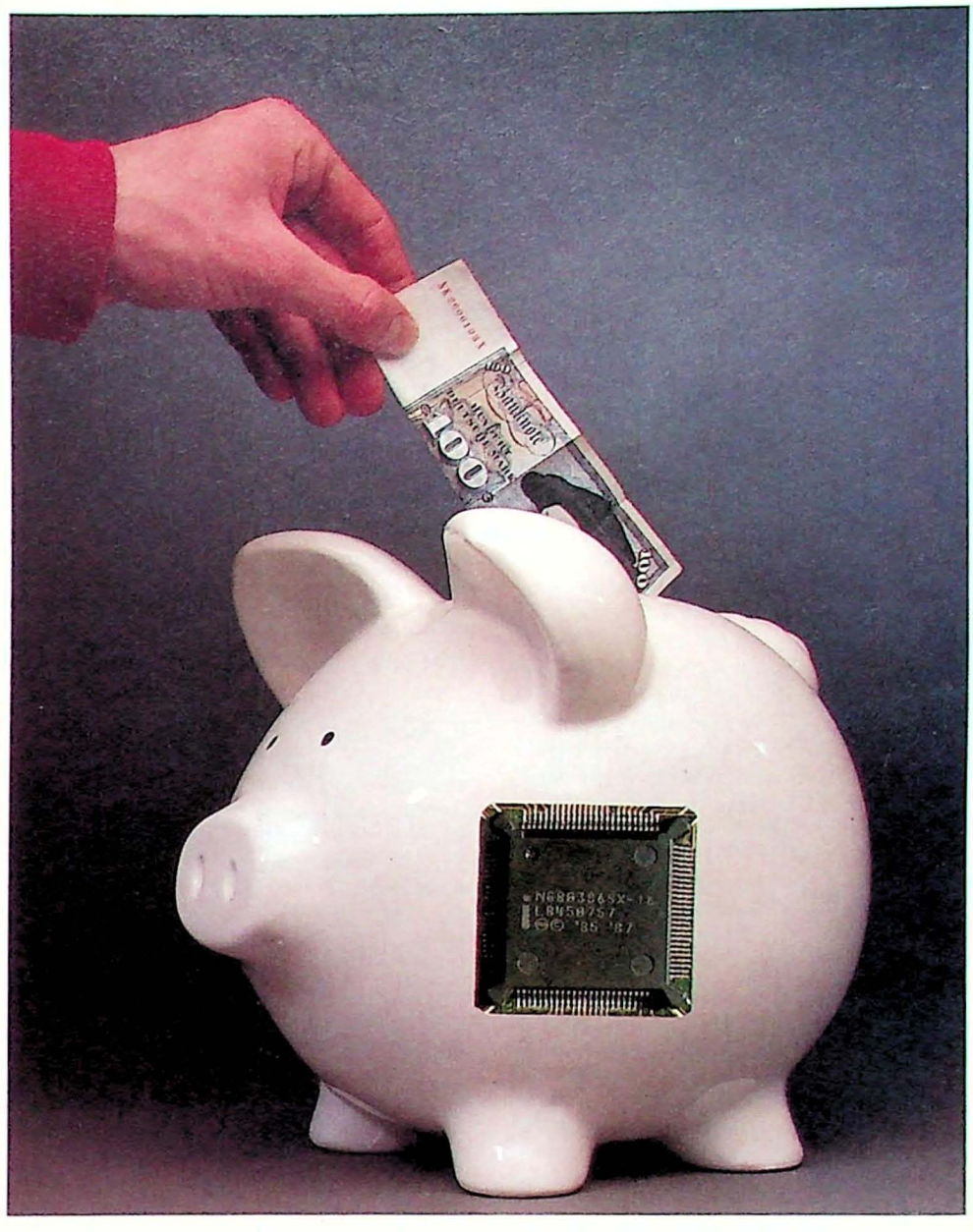
Listing 3. Die Prozedur »lese_ap.asm« zeigt ein Anwendungsbeispiel für den Zugriff auf »append«

```
[VS4]    1: ; Programmname: ALTINT25.ASM
[4E3]    2: ; (C)  Everts & Hagedorn
[8G3]    3: ;      DOS International
[P3Ø]    4: ;
[VN5]    5: ;Absolutes Lesen der Sektoren 1Ø, 11
[LG6]    6: ;und 12 von der Festplatte mit Hilfe des
[KB5]    7: ;Interrupts 25hex bei MS-DOS bis
[DX1]    8: ;Version 4.Ø
[UT3]    9: ;*********************************
[V26]   1Ø: ;Programm ist als EXE-Datei aufgebaut
[UT3]   11: ;*********************************
[P3Ø]   12: ;
[ZG4]   13: stack   segment stack 'stack'
[TZ1]   14:         db 1ØØh dup(Ø)
[PU1]   15: stack   ends
[P3Ø]   16: ;
[ØB3]   17: code    segment 'code'
[744]   18:         assume  cs:code, ds:daten
[CW2]   19: start   proc    far
[K94]   2Ø:        ;Lade Datensegment nach AX
[XC2]   21:         mov     ax,daten
[1V1]   22:         mov     ds,ax
[P14]   23:         ;lade Adresse der Meldung
[KY3]   24:         mov     dx,offset meldung
[2M1]   25:         mov     ah,9
[3K1]   26:         int     21h
[C33]   27:        ;wähle Laufwerk A
[5M1]   28:         mov     al,Ø
[KC5]   29:        ;3 Sektoren sollen gelesen werden
[GN1]   3Ø:         mov     cx,3
[UL3]   31:        ;beginne bei Sektor 1Ø
[GS1]   32:         mov     dx,1Ø
[284]   33:        ;setze Adresse des Puffers
[ZF4]   34:        ;und löse Lesevorgang aus
[BX3]   35:         mov     bx,offset tabelle
[FK1]   36:         int     25h
[815]   37:        ;Statuswort vom Stack holen
[RD1]   38:         pop     ax
[VY1]   39:         mov     ah,4ch
[3K1]   4Ø:         int     21h
[MV1]   41: start   endp
[ZP1]   42: code    ends
[P3Ø]   43: ;
[7D3]   44: daten   segment 'data'
[5H5]   45:        ;reserviere Platz für 3 Sektoren
[HN3]   46: tabelle db      512*3 dup(Ø)
[P3Ø]   47: ;
[5Ø4]   48: meldung db      Øah,Øah,Ødh
[CZ3]   49:         db      'Ich lese Sektoren '
[WC3]   5Ø:         db      'vom Laufwerk A'
[BB2]   51:         db      Øah,Ødh,'$'
[LU1]   52: daten   ends
[P3Ø]   53: ;
[BØ3]   54:        ;starte bei Start
[4W1]   55:         end     start
```

Listing 4. Das Programm »altint25.asm« zeigt den Interrupt-Aufruf 25hex bis MS-DOS 4.01

```
[4T4]    1: ; Programmname: NEUINT25.ASM
[4E3]    2: ; (C)  Everts & Hagedorn
[8G3]    3: ;      DOS International
[P3Ø]    4: ;
[DE6]    5: ;Absolutes Lesen der Sektoren 65536 bis
[3U3]    6: ;65538 von der Festplatte
[Z55]    7: ;mit Hilfe des Interrupts 25hex
[N34]    8: ;bei MS-DOS ab Version 4.Ø
[UT3]    9: ;*********************************
[V26]   1Ø: ;Programm ist als EXE-Datei aufgebaut
[UT3]   11: ;*********************************
[P3Ø]   12: ;
[ZG4]   13: stack   segment stack 'stack'
[TZ1]   14:         db 1ØØh dup(Ø)
[PU1]   15: stack   ends
[P3Ø]   16: ;
[ØB3]   17: code    segment 'code'
[744]   18:         assume  cs:code, ds:daten
[CW2]   19: start   proc    far
[F23]   2Ø:        ;Lade Datensegment
[XC2]   21:         mov     ax,daten
[1V1]   22:         mov     ds,ax
[U93]   23:        ;Gebe die Meldung aus
[KY3]   24:         mov     dx,offset meldung
[2M1]   25:         mov     ah,9
[3K1]   26:         int     21h
[FP3]   27:        ;wähle Laufwerk C aus
[7M1]   28:         mov     al,2
[WA5]   29:        ;Ein Flag zeigt neuen Aufruf an
[TF2]   3Ø:         mov     cx,Øffffh
[7V5]   31:        ;setze Adresse des Parameterblocks,
[A14]   32:        ;passe Segmentadresse im
[Ø43]   33:        ;Parameterblock an
[ZF4]   34:        ;und löse Lesevorgang aus
[9D3]   35:         mov     bx,offset para
[9V1]   36:         mov     dx,ds
[SV2]   37:         mov     paraseg,dx
[FK1]   38:         int     25h
[B33]   39:        ;Hole Flag vom Stack
[RD1]   4Ø:         pop     ax
[VY1]   41:         mov     ah,4ch
[3K1]   42:         int     21h
[MV1]   43: start   endp
[ZP1]   44: code    ends
[P3Ø]   45: ;
[7D3]   46: daten   segment 'data'
[5Ø4]   47: meldung db      Øah,Øah,Ødh
[1X4]   48:         db      'Ich lese Sektoren einer '
[RX4]   49:         db      'erweiterten Partition '
[9Z1]   5Ø:         db      Øah,Ødh
[NX3]   51:         db      'auf Festplatte C '
[243]   52:         db      'unter MS-DOS '
[D93]   53:         db      'ab Version 4.Ø'
[BB2]   54:         db      Øah,Ødh,'$'
[X33]   55:        ;Ab Sektor 1Ø lesen
[K22]   56: para    dd      1Ø
[ZG3]   57:        ;Anzahl der Sektoren
[LYØ]   58:         dw      3
[CT5]   59:        ;Segmentadresse des Datenpuffers
[643]   6Ø: paraseg dw      daten
[NE5]   61:        ;Offsetadresse des Datenpuffers
[H53]   62:         dw      offset tabelle
[5H5]   63:        ;reserviere Platz für 3 Sektoren
[HN3]   64: tabelle db      512*3 dup(Ø)
[LU1]   65: daten   ends
[P3Ø]   66: ;
[L34]   67:        ;starte bei Label START
[4W1]   68:         end     start
```

Listing 5. Das Programm »neuint25.asm« demonstriert den Interrupt-Aufruf 25hex mit MS-DOS 4.01



80386SX – Der 386-PC für jedermann

Einen 386-PC zu besitzen, wäre schon toll. Leider sind diese Geräte aber noch recht teuer. Um die Vorteile eines 386-PC für einen deutlich niedrigeren Preis verfügbar zu machen, hat Intel den Prozessor 80386SX entwickelt. Welche Vorteile er bietet und wie sie realisiert wurden, lesen Sie in diesem Beitrag.

MS-DOS auf einem 386-PC? Reine Verschwendung, sagen viele. Aber der Einsatz des 80386-Prozessors unter MS-DOS ist nicht ganz vergeblich. Schließlich hat die Software-Industrie, die sich ausgiebig mit MS-DOS-Programmen beschäftigt, seit dem Erscheinen des 80386 nicht Däumchen gedreht. Zahlreiche Programme wie Windows 386, Lotus 3.0 und Paradox 386 machen sich die Fähigkeiten des 386-Prozessors zunutze. Daß Sie diese Rechenleistung auch für weniger Geld bekommen können (Computer mit dem 386SX-Prozessor gibt es bereits ab etwa 3000 Mark), machte Intel mit der SX-Version des 80386 möglich. Damit sind die 80386SX-Computer um rund 1000 Mark billiger als ihre großen Brüder mit einem richtigen 80386. Der »alte« 386-Prozessor heißt jetzt zur besseren Unterscheidung 80386DX. Der neue, der »SX« also, ist natürlich nicht der gleiche. Alle Vorteile eines 80386DX-Prozessors bekommen Sie für das wenige Geld natürlich nicht. Der SX hat nämlich nicht wie der DX einen 32 Bit breiten Daten-Bus (Bild 1).

32 Bit oder 16 Bit: Das ist der gravierende Unterschied zwischen 386DX und 386SX

Er wurde vielmehr auf 16 Leitungen reduziert (Bild 2), ähnlich wie bei den Prozessoren 8088 und 8086. Das bedeutet praktisch, daß bei jedem 32-Bit-Zugriff eines Programms zwei Takte nötig sind, damit der Prozessor an die gewünschten Daten herankommt. Das dauert entspechend doppelt so lange. Es hat aber den Vorteil, daß die Mutterplatine des Rechners keine 32 Datenleitungen zum Übertragen der Speicherinhalte zum Prozessor und wieder zurück enthalten muß. Der Platinenaufbau ist dadurch wesentlich einfacher und natürlich auch billiger als bei einem »echten« 386-PC.
Sind die Daten erst einmal im 386SX angelangt, so rechnet und bearbeitet er sie in 32-Bit-Blöcken. Außerdem kann der 386SX aufgrund eines 24 anstatt 32 Bit breiten Adreß-Bus auch nur 16 MByte Speicher adressieren. Durch diese Konstruktion ist der 80386SX dafür geeignet, direkt auf einer AT-Hauptplatine eingesetzt zu werden. Zudem ist der SX auch noch absolut softwarekompatibel zum 386DX. Das heißt, daß Sie alle Programme verwenden können, die einen 80386 Prozessor erfordern. Sämtliche Prozessorkommandos sind bei

beiden Prozessoren gleich. Daher ist es auch schwierig, ein Testprogramm zu entwickeln, mit dem sich unterscheiden läßt, ob es sich um einen 386DX oder 386SX handelt.

Auch in seinem Äußeren hat sich der Chip verändert. Der SX (Bild 3) ist in einem preisgünstigen Kunststoffgehäuse untergebracht und nicht, wie der DX (Bild 4), in einen aufwendigen Keramikblock integriert. Dieses Kunststoffgehäuse wird direkt auf die Mutterplatine des PC geklebt und anschließend verlötet.

Am SX-Gehäuse fällt auf, daß an den vier Ecken Plastikstifte vorstehen. Das hat den Zweck, die Pins des Chips so zu schützen, daß ein Gerät zur Bestückung den Prozessor an den Kanten greifen kann, ohne die feinen Anschlüsse zu beschädigen. Ferner können die Chips so in einfacheren Verpackungen transportiert werden.

Dadurch fallen nicht nur die Kosten für den Sockel weg. Es wird auch erreicht, daß der Prozessor komplett maschinell montiert werden kann. Das macht sich dann wiederum im niedrigen Preis bemerkbar. Intel verkaufte den SX bereits im August '89 für 89 Dollar. Das ist auch der Preis für einen 16-MHz-80286 von Harris!

Durch diese Preisvorteile hofft Intel, den Markt der schnellen 80286-Prozessoren zu schwächen, die ja nicht von Intel selbst stammen.

Der SX ist von der reinen Rechenleistung her rund 10 Prozent langsamer als ein mit 16 MHz getakteter 80286-Prozessor. Daher will Intel noch Anfang 1990 eine 20-MHz-Version auf den Markt bringen, deren Geschwindigkeit die 80286-Prozessoren übertreffen soll.

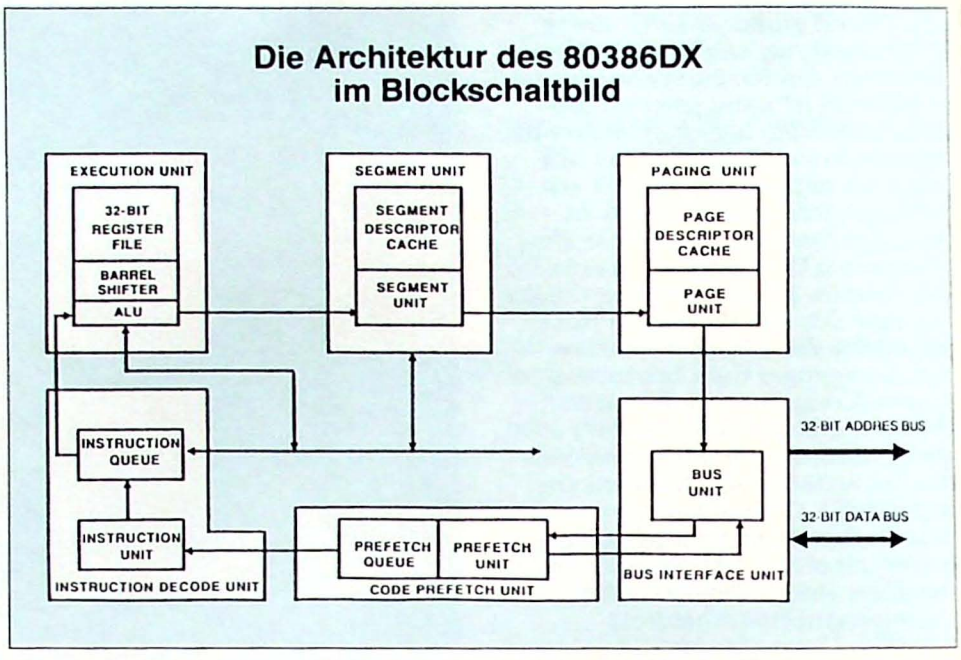


Bild 1. Der 80386DX von Intel im Blockschaltbild

Bislang liegt der entscheidende Vorteil der SX-Version des 80386-Prozessors eben nicht in der gesteigerten Geschwindigkeit, sondern in der Vielzahl der Möglichkeiten bei der Programmierung und Programmanwendung.

Ein 386-Computer – auch der SX – ist dazu in der Lage, die Hardware um ihn herum besonders gut auszunutzen. Der 80286-Prozessor kann zwar auch 16 MByte Speicher adressieren, aber nur im »Extended-Modus«.

Die wenigsten Programme nutzen diesen Umstand jedoch gewinnbringend aus. Die 80386-Prozessoren schaffen das gleiche ohne Tricks und Kniffe. Diese Art von Speicheradressierung wird dann auch entsprechend von der Software unterstützt, freilich nicht von MS-DOS. Aber einige Programme, die Sie unter MS-DOS aufrufen, schalten MS-DOS während ihrer Ausführung ab und umgehen so die Einschränkungen. Betriebssysteme, wie Windows oder Unix und deren Derivate, gibt es in speziellen 386-Versionen, die – dank der Kompatibilität beider Prozessoren – auch auf einem 386SX laufen. Dann macht sich der Unterschied zwischen 80286 und 80386 schon deutlich bemerkbar.

Gerade zum Thema Multitasking haben die beiden 386-Prozessoren einiges anzubieten. Der 80386 stellt einen sogenannten »Virtuellen Modus« zur Verfügung. Dieser Modus kann mehrere 8086-Prozessoren emulieren. Das ist etwa so, als würde aus dem einen Computer eine ganze Batterie von Computern, die voneinander unabhängig arbeiten. Das wird dadurch realisiert, daß dem Prozessor zyklisch wechselnd verschiedene Speicherbereiche zugeordnet werden. Bei jedem Wechsel des Speicherblocks simuliert der 386-Prozessor den dazugehörigen virtuellen Prozessor. Der simulierte Prozessor beherrscht natürlich auch die 32-Bit-Arithmetik.

Lassen Sie nun beispielsweise auf einem »tatsächlichen« Rechner in zwei virtuellen Rechnern verschiedene Anwendungen zur selben Zeit laufen – unter Windows oder Unix ist das möglich. Die beiden Programme kommen sich auch ohne besonderen Aufwand des Betriebssystems nicht in die Quere, wenn mehrere virtuelle Maschinen existieren, weil die eine

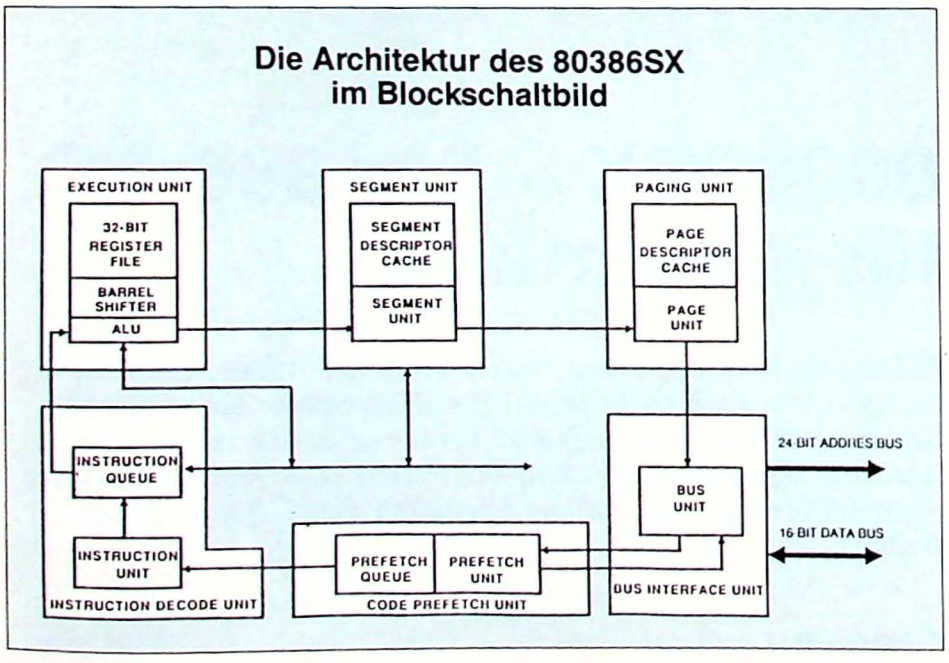


Bild 2. Im Blockschaltbild des Intel 80386SX erkennen Sie deutlich den reduzierten Daten- und Adreß-Bus

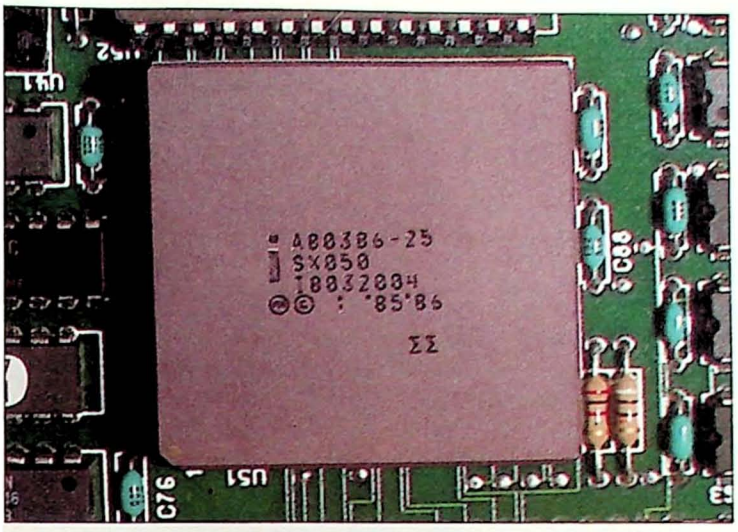


Bild 3. Das deutlich kleinere Gehäuse des 80386SX sitzt direkt auf der Mutterplatine des Rechners

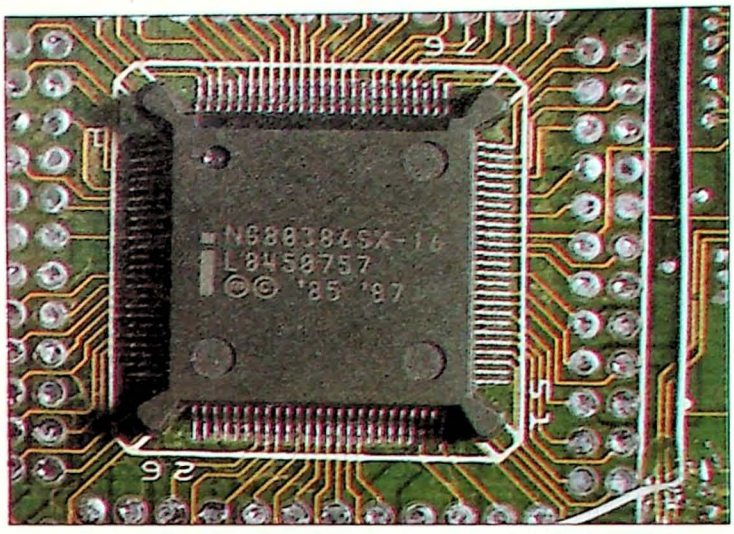


Bild 4. Umgeben von einem Keramikgehäuse sitzt der 80386DX in einem Sockel auf der Mutterplatine des PC

nicht weiß, was die andere tut. Noch viel wichtiger: Die beiden virtuellen Geräte können sich nicht ohne Hilfe des Betriebssystems beeinflussen.

Angenommen, Ihr Computersystem verfügt über 4 MByte Speicher. Starten Sie jetzt den PC mit MS-DOS, und rufen Sie anschließend Windows 386 auf. Dann starten Sie unter Windows drei verschiedene Anwendungen. Dabei wird automatisch der Speicher aufgeteilt: Das erste MByte bekommen MS-DOS und Windows, den drei Anwendungen steht dann jeweils ein weiteres MByte zur Verfügung. Alle Speicherbereiche sind fein säuberlich voneinander getrennt.

Diese Fähigkeit zum Multitasking schon auf der Hardware-Seite ist auf dem PC erst mit dem 386-Prozessor möglich geworden. Nur teuer war es leider – wenigstens bisher. Der 80386SX kann dies aber genauso gut.

Stellen Sie sich vor, Sie haben gerade ein Programm aufgerufen, das für die nächsten 20 Minuten damit beschäftigt ist, eine Adreßdatei zu sortieren. Währenddessen arbeiten Sie mit einem Textverarbeitungsprogramm ein Anschreiben aus. Das Anschreiben lassen Sie als Serienbrief ausdrucken. Damit Sie nicht vor dem Computer sitzen und sich langweilen, spielen Sie, bis die Briefe fertig sind, eine Runde Tetris oder ähnliches.

Multitasking wie in diesem Beispiel zu betreiben, setzt allerdings voraus, daß Ihr Rechner über einen entsprechend großen Arbeitsspeicher verfügt, weil den einzelnen Anwendungen Speicher zugeteilt werden soll, auf den keine andere Applikation zugreifen darf.

Auf einem 80286-Prozessor muß ein Multitasking-Betriebssystem die mangelnden Multitasking-Fähigkeiten des Prozessors selbst emulieren. Das funktioniert zumeist nicht besonders zuverlässig, und zum anderen ist diese Methode auch sehr zeitaufwendig. Dann ist ein 80386DX oder 80386SX von großem Vorteil, weil er die Geschwindigkeit eines schnellen 80286-Prozessors bei weitem überflügelt.

Des weiteren helfen Ihnen intelligente Speicherverwaltungsprogramme wie »Qemm-386« über die wenig beliebte 640-KByte-Grenze von MS-DOS hinweg. Damit zum Beispiel lagern Sie speicherresidente Hilfsprogramme vollständig in Speicherbereiche aus, die MS-DOS nicht zu adressieren vermag. Natürlich geht auch das nur mit einem 386-Prozessor.

Kurzum, der Intel-Prozessor 80386SX bietet Ihnen sämtliche Vorteile der 80386DX-Version. Die einzige Einschränkung, die Sie dabei in Kauf nehmen müssen, ist der auf 16 Bit reduzierte Daten-Bus und die damit verbundene geringere Verarbeitungsgeschwindigkeit.

Wie bereits angesprochen, ist der 80286 eins zu eins gegen den 386SX austauschbar. Wenn Sie also im Besitz eines AT sind, so eröffnet sich Ihnen nun die Möglichkeit, aus Ihrem AT einen 386-Computer zu machen – einen 386SX-PC. Eigentlich ist dies gar nicht möglich, wenn doch die SX-Prozessoren als SMD-Bauteile (Surface Mounted Device) – also als Bauteile, die direkt auf die Platine geklebt und dann verlötet werden – aufgebaut sind und somit nicht in den Sockel des 80286-Prozessor passen können.

Aber auch dieses Problem wurde schon aus der Welt geschafft (zumindest aus der amerikanischen). Einige Computerhersteller bieten nämlich sogenannte »Tochterkarten« an, auf denen ein 386SX mit der erforderlichen Elektronik zur Anpassung der nicht vollständig kompatiblen Signale sitzt. Damit Sie für diese Karte keinen Steckplatz in Ihrem PC belegen müssen, hat diese SX-Tochterkarte einen Stecker, der in den Sockel des 80286 paßt. Da stellt sich jedoch ein weiteres Problem: Den AT-Prozessor gibt es in drei verschiedenen Ausführungen, die jeweils verschiedene Sockel erfordern. Die Tochterkarten sind deshalb auch in drei verschiedenen Ausführungen erhältlich. Sie müssen nur noch herausfinden, welcher Sockel auf der Hauptplatine sitzt und welche Tochterkarte hineinpaßt.

Sofern der Prozessor in Ihrem Computer nicht gerade unter dem Netzteil sitzt oder der Platz für die Tochterkarte auf andere Weise blockiert wird, steht der Aufrüstung nichts mehr im Wege. Sie öffnen das PC-Gehäuse, entfernen den 80286-Prozessor und stecken die Tochterkarte statt seiner in den Sockel. Damit ist Ihr AT ein »386-PC«. Und das zu einem recht günstigen Preis.

Die Tochterkarten waren in den Vereinigten Staaten bereits ab August '89 zu kaufen – zu einem Preis von etwa 700 Dollar. Gar nicht so viel, wenn Sie sich überlegen, daß ein kompletter 386SX-PC mindestens 3000 Mark kostet. Noch besser wird der Preis, wenn Sie in Elektronik bewandert sind und die Tochterkarte selbst entwickeln. Der Prozessor ist schließlich schon für weniger als 100 Dollar auf dem Markt erhältlich.

Gleichgültig, ob und über welchen Weg Sie sich für den 80386SX entscheiden, aufgrund seiner klaren Vorteile und der Entschlossenheit von Intel, die 80286-Prozessoren zu verdrängen, wird der 80386SX sich in gar nicht so ferner Zukunft als der Prozessor für den AT etablieren.

(bk)

Der Draht zur Welt

Programmierung der seriellen Schnittstelle

Nichts an der Arbeit mit einem Computer ist so geheimnisumwoben wie die Kommunikation mit der Außenwelt. Kühlschränke steuern, Garagentore öffnen, Modems betreiben – all das ist für die meisten Computerbenutzer immer noch ein Buch mit sieben Siegeln.

Hatten Sie nicht auch schon den Wunsch, über Ihren PC oder AT mit der Außenwelt in Kontakt zu treten, sei es nun über ein Modem oder über die Ansteuerung eines Gerätes? Der folgende Beitrag befaßt sich deshalb mit der Ansteuerung der seriellen RS232-Schnittstelle. Es gibt verschiedene Arten von Software, mit der Sie die serielle Schnittstelle ansteuern können.

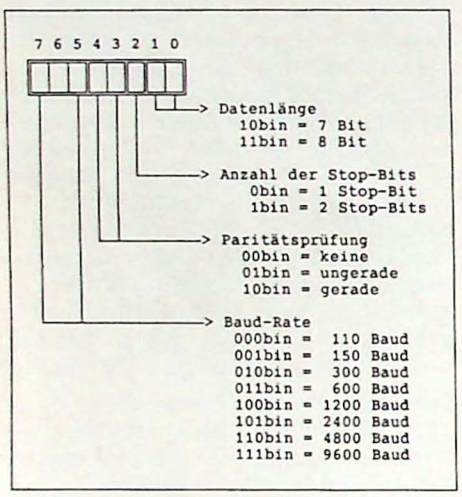


Bild 1. Protokollbeschreibung bei Aufruf der Funktion 00hex des BIOS-Interrupt 14hex

Da gibt es zum Beispiel ein Terminalprogramm (bei Modems) oder ein Datentransferprogramm (zum Beispiel Laplink). Aber auch MS-DOS bietet im BIOS vier Funktionen zur Ansteuerung der seriellen Schnittstelle an, die über den Interrupt 14hex aufgerufen werden. Das »Gehirn« einer RS232-Karte ist der »UART« (Universal Asynchronous Receiver Transmitter), ein Controller mit der Bezeichnung »8250«. Leistungsfähigere Controller wie der »16450« oder der »16550« werden für Übertragungsraten gebraucht, die größer als 9600 Baud sind. Diesem Controller müssen zuerst die benötigten Daten mitgeteilt werden, damit die Schnittstelle richtig konfiguriert werden kann. Bild 1 zeigt die Protokollbeschreibung bei Aufruf der Funktion 00hex des BIOS-Interrupt 14hex. Nach der Initialisierung liefert diese Funktion den Leitungsstatus im AH-Register zurück.

Allerdings kann nicht nur der Leitungsstatus, sondern auch der Modemstatus abgefragt werden. Die Funktion 03hex liefert den Leitungsstatus im AH- und den Modemstatus im AL-Register zurück (Bild 2). Die Bits 4 bis 7 stellen eine Wiederholung der Bits 0 bis 3 dar, denen nur das Wort »Delta« vorausgeht. »Delta« kommt aus dem Griechischen, und steht hier für Veränderung.

Das heißt, daß die Bits 0 bis 3 jeweils an zeigen, ob sich seit dem letzten Auslesen der Status der Bits 4 bis 7 verändert hat. Somit erkennen Sie beispielsweise, ob das Telefon gerade zu läuten begonnen hat oder ob es aufgehört hat zu klingeln. Am einfachsten ist die Ansteuerung über eine Hochsprache, beispielsweise Basic. Der Befehl, den Basic Ihnen zum Eröffnen und Kommunizieren über die serielle Schnittstelle zur Verfügung stellt, lautet

```
Open "COMn [Baud] [,Parität] [,Bits]
[,Stop] [Optionen]" as [#]Dateinummer
[LEN=Größe]
```

»n« steht für die Nummer des seriellen Ports (das kann 1 oder 2 sein). »Baud« ist die Konstante, mit der die Übertragungsgeschwindigkeit angegeben wird. Mögliche Werte sind hier 75, 110, 150, 300, 600, 1200, 2400, 4800 und 9600. Wird dieser Wert weggelassen, so setzt Turbo Basic den Standardwert von 300 ein. »Parität« legt das Prüfbit für die Übertragung eines Datenwortes fest und besteht aus einem einbuchstabigen Kürzel. Die zugelassenen Werte finden Sie in Tabelle 1. Als Standardwert wird hier »Even« angenommen. »Bits« ist eine Konstante im Bereich von 5 bis 8 und gibt an, aus wie vielen Bits ein Datenwort besteht. Die Standardvorgabe ist hier 7, jedoch wird heutzutage meistens der Wert 8 benutzt. Die Anzahl der Stopbits geben Sie durch »Stop« an. Erlaubt ist der Wert 1 oder 2. Außer bei 75 und 110 Baud wird dieser Wert standardmäßig auf 1 gesetzt. »Dateinummer« bezeichnet den Kanal, dem die Schnittstelle zugeordnet wird. Hier wird meist der Wert 1 angegeben, außer Sie haben gleichzeitig schon eine oder mehrere Dateien geöffnet und somit diese Kanalnummer schon vergeben. »Größe« bezeichnet die maximale Anzahl von Bytes, die mit dem Befehl »get« gelesen beziehungsweise mit »put« geschrieben werden können. Als Standardvorgabe dient hier 128. Doch Vorsicht! Dieser Wert darf die zuvor mit dem Befehl »$com« festgelegte Größe des Port-Buffers nicht überschreiten. Tabelle 2 zeigt Ihnen die Werte, die Sie unter »Optionen« noch einstellen können.

Sie öffnen also die serielle Schnittstelle beispielsweise mit

```
Open "com1:2400,n,8,1" as #1 len=1
```

Mit den beiden Befehlen »print#« oder »put« schicken Sie Zeichen an die Schnittstelle, mit »input#« oder »get« lesen Sie ankommende Zeichen.

(Thomas Wolf/wn)

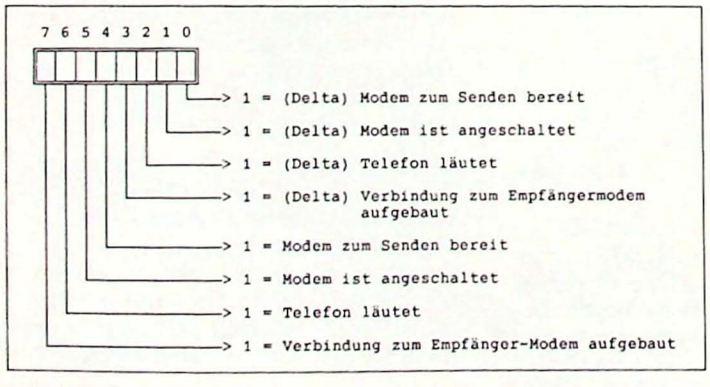


Bild 2. Der Modemstatus der seriellen Schnittstelle

Kürzel	Erklärung
S	»Space« (Paritätsbit immer »0«)
O	»Odd«, ungerade Parität
M	»Mark« (Paritätsbit immer »1«)
E	»Even«, gerade Parität
N	»None«, kein Prüfbit

Tabelle 1. Die Bedeutung der Paritätskürzel

RS	unterdrückt Signale der Leitung RTS (Request to Send)
CS[msec]	setzt die Verzögerungszeit der Leitung CTS (Clear to Send)
DS[msec]	setzt die Verzögerungszeit der Leitung DSR (Data Set Ready)
CD[msec]	setzt die Verzögerungszeit der Leitung CD (Carrier Detect)
LF	sendet hinter jedem CR (Carriage Return) ein Linefeed
PE	aktiviert die Paritätsprüfung

Tabelle 2. Die Optionen des Turbo-Basic-Befehls »open com«

MaskEdit PLUS 5.0

Professioneller Maskengenerator

mt nur Profis sind begeistert!

viele Hobbyprogrammierer haben jetzt die Vorzüge eines professionellen uktes entdeckt. Zudem ist durch die **kontextbezogene Hilfe** auf eindruck und die frei konfigurierbare Bedienung jeder in der Lage, sofort **MaskEdit Plus** konstruktiv zu arbeiten.

eern auch Sie in Sekundenschnelle **komfortable Bildschirmmasken** für i- und Untermenüsysteme, Vorder- und Hintergrundmasken usw. auf den ittor und binden Sie danach den von **MaskEdit Plus** im gewählten gabeformat erzeugten Quellcode einfach in Ihr Programm ein.

dl sparen durch Modulsystem!

h den modularen Aufbau von **MaskEdit Plus** können Sie gezielt und ennsparend einkaufen: Standard-Dateiausgabeformate wie ASCII, Binär, see und ANSI sind bereits im Grundpaket enthalten, durch **spezielle kentreiber** bedienen Sie Ihre bevorzugte Programmiersprache.

fessionelle Leistungen für jedermann!

em acht Rahmentypen sind bis zu 16 Hintergrundfarben, Schattenwurf und orkoordinaten darstellbar, und beim Rahmenzeichnen werden die **zungspunkte automatisch** angepaßt. Eine integrierte ASCII-Tabelle macht mmen mit einem **universellen Maustreiber** gehörig Tempo.

über 50 Funktionen können Sie auch **Teile einer Maske** kopieren, löschen, chieben, zentrieren, färben oder füllen. Dabei wird von **MaskEdit Plus** der pillette IBM-Zeichensatz bis 255, die gleichzeitige Bearbeitung von zwei ken und die Definition von Ein-/Ausgabefeldern unterstützt.

grammier-Highlights der Maskentreiber in Kürze:

iche Treiber werden als Sourcecode geliefert * Eigene Routinen können zur Kontrolle von Eingaben, xten, etc. problemlos in den Steuerungsablauf eingebunden werden * Integration von Datum und ler Uhrzeit und automatische Typüberprüfung bei der Eingabe, auch für den zusätzlichen Datentyp MI * Aufruf der generierten Masken durch einen einzigen Befehl und Steuerung der Eingabefelder durch den Maskentreiber * Wahlweise Ausschluß von Teilmengen des Zeichensatzes tomatische Eingabe in Großbuchstaben inklusive der Umlaute * Bei Änderungen von definierten n, kann die Kontrolle an das eigene Programm übergeben werden - Jedes Eingabefeld kann einzeln t werden, ist voll editierbar und läßt sowohl rechts- wie auch linksbündige Ausgabe zu * Bewegung Cursors innerhalb der Eingabemaske durch die Pfeil- und TAB-Tasten und Verlassen der eemaske durch Funktionstasten-Betätigung.

skEdit Plus I
o Pascal 3/4/5 und dBase III + — DM **99,– ***

skEdit Plus II
ossoft C, Quick C, Turbo C und dBase III + — DM **99,– ***

skEdit Plus III
Topspeed Modula-2 und dBase III + — DM **99,– ***

ketpreis mit zwei Treibern — DM **149,– ***

ketpreis mit drei Treibern — DM **199,– ***

monstrationsdiskette MaskEdit Plus DM **5,–**

zu passend empfehlen wir:

PERKEY 4.0
aturbelegung und Macro-Prozessor — DM **99,– ***

verbindliche Preisempfehlung. Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung.

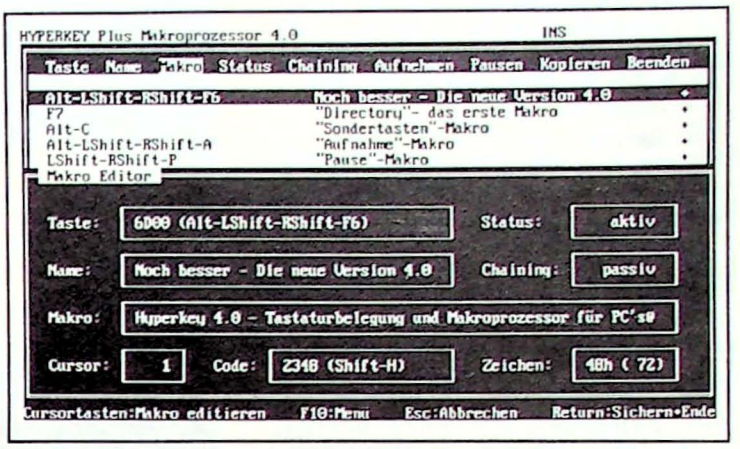
– Bitte benutzen Sie die Bestellkarte –

V-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

HYPERKEY 4.0

Geben Sie Ihren Anwendungen den letzten Schliff!
Bedienerfreundlichkeit selbstgemacht mit



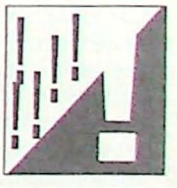
Der Tastatur-Manager

Wozu braucht man eine Tastenbelegung?
Sie können die Bedienung jedes Anwendungsprogramms wesentlich verbessern, wenn Sie wichtige Funktionen bündeln und gesammelt einer Taste Ihrer Wahl zuordnen. Oft werden bei der Programmbedienung auch F-11 und F-12 oder Kombinationen von Tasten zusammen mit SHIFT, CTRL oder ALT nicht genutzt.

Mit *HYPERKEY* können Sie alle MF-2-Tastaturen und alle Tastaturtypen von PC bis AT in Deutsch und Englisch beliebig anpassen und belegen, wobei Ihnen bis zu 500 Tastenkombinationen zur Verfügung stehen, da HYPERKEY auch zwischen SHIFT-links und SHIFT-rechts unterscheidet.

Und wenn meine Anwendung bereits Tastenbelegung vorsieht?
Anwenderprogramme können oder wollen Ihnen aus Platzgründen nicht alle technischen Möglichkeiten des Tastaturmanagements bieten. Zudem müssen einmal gefundene Belegungen in jedem Programm immer wieder an die wechselnden Anforderungen Ihrer Praxis angepaßt werden.

Mit *HYPERKEY* integrieren Sie alle Ihre Anwendungs-Belegungen in einem Programm, können mit kontextsensitiver Hilfestellung Ihre Belegungen komfortabel editieren, abspielen, speichern, sortieren, packen, verketten und suchen und damit systematisch die Bedienung aller Ihrer Anwendungen optimieren.



HYPERKEY 4.0
– der Tastaturmanager –

für alle PC/XT/AT unter MS-DOS mit minimal 256 kByte RAM, DOS ab Version 2.11 bis 3.3. Die Nutzung des Funktionsumfangs von HYPERKEY ist unter GEM von Digital Research und Microsoft Word 4/5 eingeschränkt, unter Microsoft Windows nicht, aber bei Anwendungen unter Microsoft Windows vollständig möglich.

HYPERKEY 4.0, Makroprozessor — DM **99,– ***

dazu passend empfehlen wir

RSM - Resident Software Manager — DM **69,– ***

* Unverbindliche Preisempfehlung. Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung.

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Festplatteneinbau von A bis Z

Immer wieder werden in Anzeigen Festplatten angepriesen. Haben Sie solch ein hochempfindliches Gerät dann gekauft, müssen Sie feststellen, daß die beigelegte Dokumentation den Einbau meist nur unzureichend beschreibt. Daß damit dann eine Menge Ärger verbunden ist, können Sie sich leicht vorstellen. Damit Ihnen das erspart bleibt, ist im folgenden der Einbau eines Festplattenlaufwerks vom Anfang bis zum Ende beschrieben.

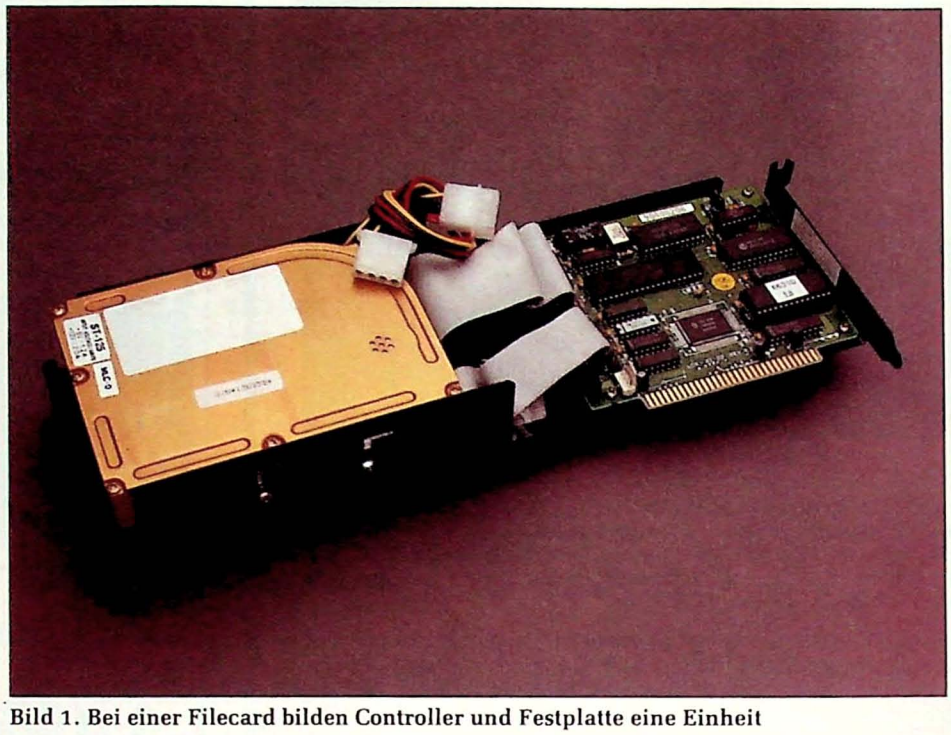


Bild 1. Bei einer Filecard bilden Controller und Festplatte eine Einheit

Festplatten gibt es viele. Welche für Sie die richtige ist, hängt von Ihren Erfordernissen, dem vorhandenen PC und nicht zuletzt von Ihrem Geldbeutel ab. Das beste Preis-Leistungs-Verhältnis finden Sie bei den Festplatten mit etwa 40 MByte Speicherkapazität. Diese Größe ist in der Regel ausreichend, um die nötigen Anwendungen und Daten aufzunehmen. Außerdem sind Festplatten mit dieser Speicherkapazität deutlich preiswerter als Geräte mit 60, 80 oder mehr MBytes Fassungsvermögen. Dann spielt natürlich auch noch die Geschwindigkeit der Festplatte eine große Rolle.

Die Geschwindigkeit, mit der die Daten und Programme in den Arbeitsspeicher Ihres PC geladen werden, hängt zum einen von der Zugriffszeit der Festplatte und zum anderen von der Geschwindigkeit des verwendeten Controllers ab. Ein schneller Controller ist sogar wichtiger als eine schnelle Festplatte. Die Zugriffszeit der Festplatte kann noch so kurz sein – wenn der Controller die Daten nicht schnell genug aufnehmen und an den PC weiterleiten kann, müssen Sie trotzdem längere Ladezeiten in Kauf nehmen.

Nach diesen Kriterien haben Sie sich nun für eine Festplatte entschieden und wollen sie auch in Ihren PC einbauen – und zwar so, daß sie funktioniert. Dazu müssen Sie das hochempfindliche Gerät zuerst einmal auspacken. Gehen Sie dabei sehr vorsichtig zu Werke, weil Sie nie sicher sein können, daß der Hersteller die Schreib-/Leseköpfe in einer sicheren Position geparkt hat. Wenn er das nicht getan hat und die Festplatte erschüttert wird, kann dieser Umstand dazu führen, daß die Köpfe auf die Platten stoßen. Dabei kann sehr leicht die empfindliche Plattenoberfläche beschädigt werden. Deshalb müssen Sie auch beim Transport der Festplatte sehr behutsam vorgehen. Sonst wäre das teure Gerät schon Opfer eines sogenannten »Headcrash« geworden, bevor es überhaupt zum Einsatz kommen kann.

Wenn Sie sich eine Filecard gekauft haben, ist die Sache relativ einfach. Eine Filecard ist eine Festplatte, die auf einer Steckkarte zusammen mit dem Controller montiert ist (Bild 1). Das hat den Vorteil, daß der Einbau recht schnell vonstatten geht. Den Einbau von Filecards nehmen Sie genau so vor, wie im folgenden für die Controller-Karte beschrieben. Beachten Sie aber, daß die meisten Filecards so seltsam konstruiert sind, daß die Festplatte auf der Platine über den benachbarten Steckplatz ragt und so verhindert, daß direkt nebenan eine andere Steckkarte eingebaut werden kann. Um dieses Manko zu umgehen, stecken Sie die Filecard einfach in den äußersten Steckplatz, so daß die Platte nur über die Hauptplatine ragt, alle übrigen Steckplätze aber für spätere Aufrüstungen oder Erweiterungen frei zugänglich bleiben.

Bei »richtigen« Festplatten wird es schon komplizierter. Wenn Sie eine 3 1/2-Zoll-Festplatte verwenden möchten, aber nur 5 1/4-Zoll-Einschübe in Ihrem PC besitzen, benötigen Sie einen Einbaurahmen. Dieser Einbaurahmen verbreitert Ihre 3 1/2-Zoll-Festplatte so, daß sie in die normalen 5 1/4-Zoll-Einschübe des Computers paßt. Leider wird dieses nützliche Zubehör nur selten mitgeliefert. Sie können aber so einen Einbaurahmen für etwa 30 Mark erwerben.

Wenn zu der Festplatte und der meist wenig aussagekräftigen Dokumentation noch vier Schrauben mitgeliefert werden, können Sie schon zufrieden sein. Werden ein Paar Einschubleisten und ein zwanzigpoliges Flachbandkabel mitgeliefert, ist das bereits das höchste der Gefühle. Gehören diese Utensilien nicht zum Lieferumfang, bleibt Ihnen nichts anderes übrig, als sie selbst zu besorgen. An Werkzeug benötigen Sie für den Einbau nur einen Kreuzschlitzschraubendreher.

Oft kann der Einbau von Festplatten nur mit Zubehör vorgenommen werden, das selten mitgeliefert wird

Entfernen Sie jetzt alle Kabel von Ihrem Computer. Lösen Sie dann alle Befestigungsschrauben an der Rückseite des Computergehäuses. Wenn Sie jetzt den Gehäusedeckel entfernen, achten Sie darauf, daß er nicht an Bauteilen wie zum Beispiel Diskettenlaufwerken oder Steckkarten hängenbleibt. Sobald Sie den Deckel geöffnet haben, suchen Sie einen eventuell vorhandenen Controller. Den Controller erkennen Sie daran, daß von ihm ein Flachbandkabel mit 34 Adern direkt zum Diskettenlauf-

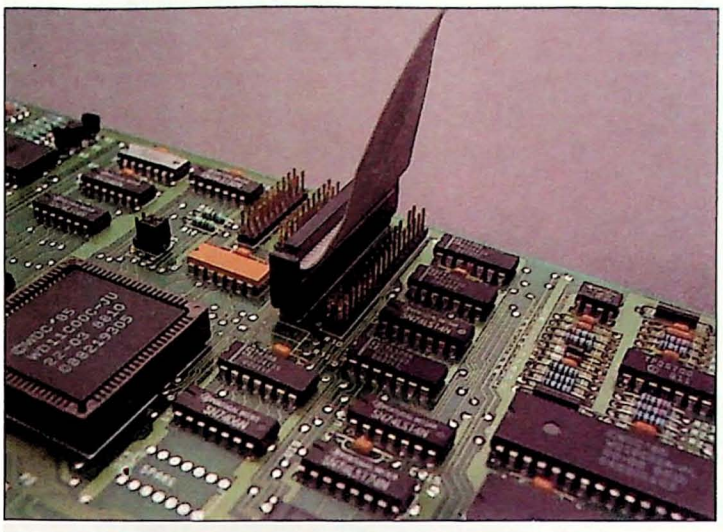


Bild 2. Ein Kombi-Controller, der zwei Disketten und zwei Festplatten verwalten kann

werk führt. Wenn sich auf der Controller-Karte eine unbelegte 34polige Steckerleiste mit der Bezeichnung »HDD«, »Winchester« oder ähnlichem befindet und außerdem noch eine 20polige Steckerleiste, die ebenfalls nicht belegt ist, handelt es sich um einen »Kombi-Controller«. Ein Kombi-Controller (Bild 2) kann neben zwei Diskettenlaufwerken auch noch zwei Festplatten ansteuern. Das heißt, dieser Controller ist für Ihre Zwecke geeignet.

Sollten Sie solche Anschlüsse nicht finden können, so handelt es sich nicht um einen Kombi-Controller, und Sie müssen entweder einen zweiten Controller einbauen oder ihn durch einen Kombi-Controller ersetzen. Die zweite Methode hat den Vorteil, daß Sie damit einen Steckplatz weniger belegen. Dafür ist ein Kombi-Controller auch etwas teurer. (Ein »Nur-Festplatten-Controller« kostet etwa 120 bis 200, ein Kombi-Controller zwischen 250 und 500 Mark)

Controller ist nicht gleich Controller: Ein Kombi-Controller ist meistens die beste Lösung

Die Controller sind in der Regel bereits durch den Hersteller richtig konfiguriert. Aber sicher ist sicher: Überprüfen Sie die Positionen der »Jumper« auf der Controller-Karte. Jumper sind winzige Stecker, mit denen Kontaktstifte auf Platinen überbrückt werden, um Konfigurationen vornehmen zu können. Wie die Jumper stecken sollen, entnehmen Sie der Beschreibung Ihrer Controller-Karte.

Für den Einbau des Controllers suchen Sie sich einen freien Steckplatz in Ihrem PC und setzen ihn dort ein, jedoch nicht bevor Sie das Abdeckblech an der Rückseite den Gehäuses entfernt haben. Solche Abdeckbleche finden Sie hinter jedem Steckplatz. Diese Bleche decken die Öffnung in der Rückseite des Computergehäuses ab und sind mit einer Schraube an der oberen Kante befestigt. Lösen Sie hierzu die entsprechende Schraube, und ziehen Sie das Blech senkrecht nach oben aus dem Computer. Wählen Sie den Steckplatz so, daß die Platine möglichst weit von den anderen Steckkarten entfernt ist. So ist besser für den Wärmeaustausch im PC gesorgt. Wenn Sie das erledigt haben, befestigen Sie das Abdeckblech der Steckkarte mit einer Schraube an der Gehäuserückseite.

Jetzt dürfen Sie sich der Festplatte selbst zuwenden. Stecken Sie das breite und das schmale Flachbandkabel mit den passenden Steckern auf der Controller-Karte zusammen. Achten Sie dabei auf die richtige Polung. Die beiden Flachbandkabel sind an einer Seite markiert. Es handelt sich um eine besonders gefärbte Leitung. Die Stecker der beiden Kabel haben auf dieser Seite den Anschluß »1«. Der dazugehörige »Pin 1« ist auf der Controller-Karte ebenfalls gekennzeichnet. Stecken Sie das Kabel also dementsprechend auf die Controller-Karte.

Besitzt Ihr PC Laufwerksschächte, die für den Festplatteneinbau mittels Einschubleisten geeignet sind, schrauben Sie nun die Einschubleisten an die beiden Längsseiten der Festplatte.

Jetzt stecken Sie Ihre neue Festplatte in einen freien Laufwerksschacht. Dazu müssen Sie aber vorher die Frontblende entfernen. Das ist eine Plastikplatte, die den Einschub abdeckt, solange er nicht verwendet wird. Bevor Sie die Festplatte im Laufwerksschacht festschrauben, stecken Sie die anderen Enden der Flachbandkabel, die Sie vorher an den Controller angesteckt haben, an die Festplattenplatine. Hier orientieren Sie sich zunächst an dem breiten Kabel. Der breite Stecker an der Festplattenplatine hat einen kleinen Schlitz. Auf der einen Seite des Schlitzes befinden sich zwei, auf der anderen Seite 15 Anschlüsse. (Die gleiche Anzahl von Anschlüssen befindet sich auch auf der anderen Seite der Platine, was eine Gesamtzahl von vier und 30 Anschlüssen ergibt.) Stecken Sie den Stecker des Flachbandkabels so auf die Platine, daß die markierte Seite des Kabels auf die Seite mit den zwei beziehungsweise vier Anschlüssen kommt. Den schmalen Stecker verbinden Sie so, daß das Kabel auf der gleichen Seite aus dem Stecker herauskommt wie bei dem breiten Kabel. Meistens geschieht das in Richtung des Festplattengehäuses. Oft sind die Stecker an der Festplattenseite des Controller-Kabels mit einem mechanischen Verpolungsschutz versehen. Das ist ein kleiner Steg im Stecker, der in den Schlitz auf der Festplattenplatine paßt, so daß Sie das Kabel gar nicht falsch herum anschließen können.

Nicht immer sind die Flachbandkabel vom Controller zur Festplatte mit einem Verpolungsschutz versehen

Anschließend müssen Sie eines der vierpoligen Kabel des Netzteils mit der Festplatte verbinden. Achten Sie auch hier unbedingt auf die richtige Polung. Diese Steckverbindungen sind immer gegen eine Verpolung geschützt. Dieser Verpolungsschutz ist ebenfalls mechanisch, besteht also nicht aus einer speziellen Elektronik, sondern aus der Konstruktion der Stecker. Die Stecker können aufgrund ihrer Form nur richtig herum eingesteckt werden. Sollte in Ihrem PC kein Netzteilkabel mehr frei sein, so benötigen Sie einen sogenannten »Y-Adapter«, mit dem Sie den Strom von einem bereits belegten Stecker abzweigen.

Sofern Ihr Computer ein Metallgehäuse hat, sind an einigen Stellen mehrere, meist schwarze Massekabel am Gehäuse festgeschraubt. Am Festplattengehäuse befindet sich eine Metallzunge, mit der Sie eines der Kabel verbinden. Gewöhnlich werden hier einfache Kabelsteckverbindungen, wie man sie auch in Autos findet, eingesetzt.

Sollten Sie kein Massekabel finden können, oder sollte Ihre Festplatte keinen Anschluß hierfür haben, ist

das nicht weiter schlimm, weil an dem Netzteilanschluß ohnehin eine Masseleitung vorhanden ist. Jetzt haben Sie es geschafft – Ihre Festplatte ist fertig eingebaut. Sie können das Gehäuse beruhigt zuschrauben.

Schließen Sie nun Ihren Computer an, und schalten Sie ihn ein. Wenn die Festplatte korrekt läuft und der Computer von der Diskette bootet, dann sind alle Kabel richtig angeschlossen. Jetzt gilt es dem Computer mitzuteilen, daß er eine neue Festplatte hat. Bei einem AT oder 386-PC ist es dazu erforderlich, daß Sie die CMOS-Konfiguration aktualisieren. Im Setup gibt es einen Punkt, der Festplatten betrifft. Dort stellen Sie entsprechend den technischen Daten der Festplatte den richtigen Typ ein.

Zudem müssen Sie die Festplatte zur Benutzung mit MS-DOS vorbereiten. Verwenden Sie dafür entweder ein Programm wie den »Disk Manager«, der im Lieferumfang von Seagate-Festplatten enthalten ist. Viele Fachhändler liefern das Programm »Speed Stor« mit (Bild 3). Diese Programme sind fast alle menügeführt und recht komfortabel. Wie Sie im einzelnen mit diesen Programmen vorgehen, schlagen Sie im Handbuch des Programms nach.

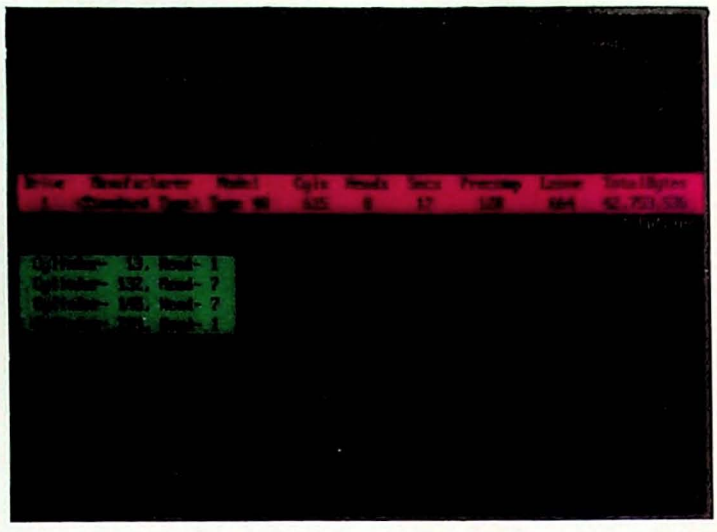


Bild 3. Das Programm »Speed Stor« findet selbständig Defekte auf der Festplatte

Im BIOS der meisten Festplatten-Controller befindet sich eine Routine für das »Low-Level-Format« der Festplatte

Wenn bei Ihrer Festplatte keine Software mitgeliefert wurde, müssen Sie mit der im BIOS des Festplatten-Controllers eingebauten Routine (nur bei PCs und Filecards) vorliebnehmen. Auch diese Routine ist meistens menügesteuert. Nur eben nicht so komfortabel.

Um die BIOS-Routine zu aktivieren, legen Sie eine MS-DOS-Diskette in Ihr Laufwerk und schalten den Computer ein. Er wird nun wie gewöhnlich starten und das Betriebssystem laden.

Es gibt einige Controller, die den Einsprung in die entsprechende Routine nach dem ersten Einschalten des Computers selbst vornehmen. Gehört Ihr Controller zu dieser Kategorie, werden Sie eine entsprechende Meldung auf Ihrem Bildschirm lesen können. Sollte der Controller den Einsprung nicht automatisch tätigen, so müssen Sie die Vorformatierungsroutine mit Hilfe des Programms »Debug« selbst aufrufen.

Hierzu rufen Sie zuerst Debug auf. Es meldet sich mit einem Minuszeichen am linken Rand des Bildschirms und wartet auf Ihre Eingaben. Der korrekte Befehl zum Aufrufen einer Routine lautet »g = «. Dahinter müssen Sie noch die richtige Startadresse der BIOS-Routine in Hexadezimalzahlen angeben. Diese Adresse ist von der Ausführung Ihres Controller-BIOS abhängig. Meistens handelt es sich um eine der folgenden drei Adressen. Demnach lautet der vollständige Startbefehl für die Vorformatierung:

```
g=c800:5
g=c800:6
```

oder

```
g=c800:1000
```

Normalerweise ist die richtige Adresse in der Anleitung zu Ihrem Festplatten-Controller vermerkt. Sollten Sie sie dennoch nicht finden, müssen Sie diese drei Möglichkeiten durchprobieren, bis sich die Routine auf dem Monitor meldet. Das ist etwas zeitaufwendig, weil der PC bei einer falschen Einsprungadresse fast immer abstürzt. Wenn es sich nicht um eine dieser drei angegebenen Adressen handelt, muß die korrekte Adresse in der Controller-Beschreibung vermerkt sein.

Auch die BIOS-Routine verlangt einige Eingaben. Entnehmen Sie die entsprechenden Daten Ihrer Anleitung, weil sie vom jeweiligen Festplattentyp abhängig ist. Sie werden unter anderem gefragt, mit welchem Interleave-Faktor Sie die Festplatten formatieren möchten. Der kleinstmögliche Interleave-Faktor hängt vom Controller ab.

Mit »Fdisk« richten Sie physikalische Abschnitte auf Ihrer Festplatte ein, die dann logischen Laufwerken zugeordnet werden

Welchen Sie einstellen müssen, steht in Ihrer Controller-Beschreibung. Je niedriger der Faktor ist, um so schneller werden später die Daten von der Festplatte in den Arbeitsspeicher des Rechners geladen. Entsprechend sind auch die Controller mit dem besten Interleave-Faktor von 1:1 am teuersten.

Die Tatsache, daß Ihr Controller die Vorformatierung der Festplatte begonnen hat, erkennen Sie leicht daran, daß die Kontrollampe der Platte leuchtet oder blinkt. Die Routine wird voraussichtlich ihre Arbeit nach spätestens fünf Minuten – abhängig von der Größe der Festplatte – beendet haben. Im Anschluß daran müssen Sie auf der Festplatte die Partitio-

```
C:\>format c:

ACHTUNG! ALLE DATEN IN DEM
PLATTENLAUFWERK C: WERDEN GELÖSCHT!
Formatierung fortsetzen?  (J/N): j

Formatierung beendet!

  33554432     Bytes Gesamt-Disketten-/Plattenspeicher
     24576     Bytes in fehlerhaften Sektoren
  33529856     Bytes auf Diskette/Platte verfügbar


C:\>
```

Bild 4. Dieser Anblick bietet sich Ihnen, nachdem Sie letzte Hand angelegt haben

nen, das heißt die physikalischen Abschnitte, einrichten. Hierbei wird die Festplatte in ein oder mehrere logische Laufwerke unterteilt. Dies läßt sich ebenfalls mit Hilfe der oben genannten Programme durchführen. Wenn bei Ihrer Festplatte keines mitgeliefert wurde, ist das MS-DOS-Programm »Fdisk« die einzige Möglichkeit.

Das Programm Fdisk ist ebenfalls menügesteuert. Wie es genau funktioniert, lesen Sie im MS-DOS-Handbuch nach. Fdisk wird Sie unter anderem fragen, wie groß Sie die Partitionen gerne hätten. Dabei müssen Sie beachten, daß unter MS-DOS erst ab Version 4.0 größere Partitionen als 32 MByte möglich sind. Aber auch bei Versionen vor 4.0 ist es nicht immer erforderlich, eine Festplatte in mehrere Partitionen zu unterteilen. Es gibt dazu einige Treiberprogramme, die Sie in die Config-Datei einbinden müssen. Sie umgehen dann die lästige 32-MByte-Grenze. Leider werfen diese Treiberprogramme bei einigen wenigen Programmen Probleme auf. Das tut MS-DOS 4.0 allerdings auch, was auf die mangelnde Kompatibilität der neuen Dateiverwaltung zurückzuführen ist. Zurück zu Fdisk: Der Computer greift jetzt kurz auf die Festplatte zu und fordert Sie dann auf, das System neu zu starten. Das geschieht, sobald Sie eine beliebige Taste drücken. Der Computer initialisiert sich jetzt neu, und Sie dürfen mit der Formatierung Ihrer Festplatte fortfahren.

Große Festplatten mit mehr als 32 MByte müssen Sie in »logische Laufwerke« partitionieren

Es hängt nun von Ihren Wünschen und von der Konfiguration Ihres Computers ab, ob Sie auf der Festplatte ein Betriebssystem einrichten wollen oder nicht. Wenn die Festplatte, die Sie eben eingebaut haben, die einzige im System ist, richten Sie am besten ein Betriebssystem auf ihr ein. Geben Sie dazu den Befehl

```
format c: /s
```

ein. Jetzt wird Ihre Festplatte endgültig für den Gebrauch formatiert. Im Anschluß daran überträgt der Computer die MS-DOS-Systemdateien von der Diskette auf die Festplatte. Wenn Sie die Option »/s« des Format-Befehls nicht angeben, wird die Festplatte formatiert, ohne daß das Betriebssystem übertragen wird. Achten Sie darauf, daß sich in Laufwerk A eine Diskette befindet, die das Betriebssystem enthält. Es steht Ihnen frei, mit dem Format-Befehl auch noch die Option »/v« anzugeben. Dieser Parameter veranlaßt den Computer, Sie nach beendeter Formatierung zu fragen, welchen Namen Sie Ihrer Platte geben wollen. Sollte Ihnen später einmal der Name nicht mehr gefallen, oder möchten Sie Ihre Platte erst später taufen, ist das kein Problem. Setzen Sie in diesem Fall den Befehl

```
label c:
```

ein, und schon bekommt Ihr Kind einen neuen Namen.

Wenn Sie nun die Formatierung der Festplatte gestartet haben, müssen Sie nicht unbedingt auf den Computer warten. Gehen Sie ruhig einen Kaffee trinken oder mit Ihrem Hund spazieren, der Rechner ist einige Zeit beschäftigt und dabei nicht auf Ihre Hilfe angewiesen. Sobald der Computer fertig ist, teilt er es Ihnen über den Monitor mit und meldet die Zahl der verfügbaren Bytes auf der Festplatte – ein (hoffentlich) erfreulicher Anblick (Bild 4).

Es kommt manchmal vor, daß Ihr Computer es nicht schafft, das Betriebssystem auf die Festplatte zu übertragen. Ist es bei Ihnen so geschehen, versuchen Sie die Systemübertragung nachträglich mit

```
sys c:
```

und

```
copy command.com c:\
```

vorzunehmen. Wenn das auch nicht hilft, bleibt Ihnen keine andere Wahl, als die Formatierung neu zu starten – Ihr Hund wird es Ihnen danken.

Sobald Sie das alles erfolgreich bewältigt haben, können Sie beginnen, Ihre Daten auf die Festplatte zu kopieren. Überlegen Sie sich aber, wie Sie Ihre neue Festplatte aufgliedern. Eine Festplatte ist so groß, daß Sie in keinem Fall alle Dateien im Hauptverzeichnis unterbringen können. Zu schnell würden Sie den Überblick verlieren. Am besten ist es, alle Betriebssystembefehle in ein eigenes Unterverzeichnis zu kopieren.

Um ein Unterverzeichnis (Subdirectory) anzulegen, verwenden Sie den Befehl »md«, was soviel bedeutet wie »make directory«. Tippen Sie also

```
md dos
```

ein, und Ihr Computer richtet ein Unterverzeichnis mit dem Namen »dos« ein. Zum Wechseln der Verzeichnisse gibt es den Befehl »cd« (change directory). In dieses Verzeichnis wechseln Sie mit dem Befehl

```
cd dos
```

Jetzt kopieren Sie alle Systemdateien, die sich auf Ihren MS-DOS-Disketten befinden, in dieses Unterverzeichnis. Geben Sie hierzu einfach den Befehl

```
copy a:*.*
```

ein. Auf dem Monitor sehen Sie jetzt jede einzelne Datei, die kopiert wird.

Natürlich wollen Sie diese »externen« Befehle (externe Befehle sind Systembefehle, die nicht in »command.com« enthalten sind, sondern in einer eigenen Datei, wie zum Beispiel »Sort« oder »Find«) auch verwenden können, ohne jedesmal den Pfad vorher angeben zu müssen. Dies erreichen Sie mit dem Befehl »path«. Es ist am sinnvollsten, wenn Sie diesen Befehl in Ihre Autoexec-Datei schreiben. Die vollständige Anweisung lautet

```
path c:\dos
```

Weitere Verzeichnisse, in denen der PC nach dem aufgerufenen Programm suchen soll, geben Sie durch ein Semikolon getrennt dahinter an (zum Beispiel »path c:\dos;c:\tools;d:\word«). Der Computer wird dadurch veranlaßt, ein Programm, das Sie aufrufen, nicht nur in Ihrem aktuellen Verzeichnis zu suchen, sondern auch in den Unterverzeichnissen »dos«, »tools« und »word«.

Unterverzeichnisse sorgen für einen besseren Überblick auf Ihrer Festplatte

Damit sind der Festplatteneinbau und die softwaremäßige Installation beendet.

Nachdem Sie nun im Schweiße Ihres Angesichts Ihren Computer um eine Festplatte bereichert haben, möchten Sie diese sicherlich auch vor Beschädigungen schützen. Schließlich sind Daten und hochempfindliche Peripherie wie Festplatten wertvoll und teuer. Gemeint ist das Sichern der Schreib-/Leseköpfe beim Ausschalten des Computers. Es gibt einige Festplatten, die ihre Schreib-/Leseköpfe beim Ausschalten automatisch »parken«, das heißt, die Köpfe werden über einen Bereich der Festplatte bewegt, der keine Daten enthält. Die etwas besseren Versionen von Autopark-Festplatten heben sogar die Schreib-/Leseköpfe von der Plattenoberfläche ab und verriegeln sie. Das nennt man »Headlift« oder »Headlock«.

Alle anderen Festplatten müssen Sie dagegen mit Hilfe eines Programms »parken«. Diese Programme heißen »park.com«, »ship.com« etc. und sollten im Lieferumfang der Festplatte enthalten sein. Leider fehlen solche Hilfsprogramme allzu oft. Manche Controller haben eine Parkroutine in ihrem BIOS integriert. Sie wird genauso aktiviert wie die »Low-Level-Format-Routine«. Diese Vorsichtsmaßnahme schützt Ihre Daten und die Festplatte sicher vor einem Headcrash.

(Bernhard Hecker/bk)

Dem PC auf den Zahn gefühlt

Wer die Produktanzeigen in Computerfachzeitschriften, die Werbeplakate beim Fachhändler und eventuell noch dessen Aussagen kritisch aufnimmt, stellt zumindest eine Gemeinsamkeit fest: Mit Superlativen wird nicht gegeizt. Doch wie hoch ist die Übertragungsgeschwindigkeit der Grafikkarte wirklich, oder womit ist der PC tatsächlich ausgestattet? Fragen, die das Programm »config.pas« nicht offenläßt.

```
PC-Config V2.25  (C) DOS International  Autor: Michael Holin

Dieser Rechner ist ein.....: AT
BIOS vom 04/30/87..........: (c) COPYRIGHT 1984,1987 Award Software Inc
Relative Geschw. zum IBM-PC: 517%      CPU: 80286, 10.0 MHz, 1 Waitstate(s)
BIOS-Erweit. gefunden bei..: C000hex
Arbeitsspeichergröße.......: 640 KByte, davon sind belegt: 112 KByte
Extended Memory............: 384 KByte frei
Installierte Grafikkarte...: VGA mit 8-Bit-Bus, 256 KByte und Farbmonitor
VideoRAM-Durchsatz.........: 166 KByte/s, Scroll: 12,8 ms, 11 VideoWaits
Zeichenausgabe BIOS, DOS...: 3398 Byte/s, 1746 Byte/s
Momentaner Videomodus......: 3, 80*25 Zeichen, farbig
DOS-Version................: 3.30
Mathematischer Coprozessor.: Nein
Anzahl Diskettenlaufwerke..: 1
Serielle, parallele Ports..: 1, 1
Kombi-Controller vorhanden.: Ja
Daten für Festplatte 1 ....: 42,8 MByte, 6 Köpfe, 819 Zylinder, 17 Sekt/Zyl
Maximaler Daten-Durchsatz..: 252 KByte/s  ==>  Wahrsch. Interleave: 2
Durchschnittl. Zugriffszeit: 30 ms,  Spur zu Spur: 7 ms

C:\>
```

Das Progamm »config.pas« zeigt, was in Ihrem PC steckt

Mit rasanter Geschwindigkeit wird die Entwicklung im Computerbereich vorangetrieben, und mit immer beeindruckenderen Leistungsdaten sieht sich der Anwender konfrontiert. Daher wird der Wunsch laut nach einer einfachen Methode, die Ausstattung und vergleichbare beziehungsweise zum Vergleich geeignete Daten bei Computern zu ermitteln.

Das Programm »config.pas« (Listing) stellt gewissermaßen eine Mischung aus »Landmark«, »SI« und »Coretest« dar. Es bestimmt all die Werte eines Computers, die etwas über seine Leistung im Vergleich zu anderen Rechnern aussagen (Bild):

- die Größe des Arbeitsspeichers
- eine eventuell vorhandene Speichererweiterung
- den Prozessortyp und dessen Taktfrequenz
- Anzahl und Art der vorhandenen Schnittstellen
- die Speicherkapazität der Festplatte.

Um einen einfachen Vergleich mit anderen Rechnern zu ermöglichen (Tabelle), wird weiterhin die Rechengeschwindigkeit in Relation zum IBM-PC angegeben (über die Aussagefähigkeit eines solchen Wertes gibt es nach wie vor geteilte Meinungen). Dieser Wert ist direkt abhängig vom Prozessortyp, der Taktfrequenz und den Waitstates und bietet einen guten Anhaltspunkt, um die reine Rechengeschwindigkeit einzuschätzen. (Außerdem stimmt der Wert ziemlich genau mit dem des Landmark-Speed-Tests überein.)

Der »Video-RAM-Durchsatz« gibt die ungefähre Menge von Zeichen an, die sich pro Zeiteinheit direkt in den Bildschirmspeicher schreiben lassen. Dieser Wert und die Zeit, die benötigt wird, den Bildschirminhalt eine Zeile nach oben zu schieben (Scroll-Vorgang), hängen hauptsächlich von der Geschwindigkeit der Grafikkarte ab. Aber auch die Geschwindigkeit des Prozessors (»CPU«) beeinflußt diese Werte – ebenso wie sie die Anzahl der Video-Waitstates (»Video-Waits«) bestimmt. Dabei dienen letztere nur als Anhaltswert beim Vergleich mit anderen Grafikkarten. Diese Angabe ist daher keinesfalls als Absolutwert zu betrachten. Es gilt allerdings allgemein die Regel: je weniger Video-Waits, desto schneller die Grafikkarte.

Der Datendurchsatz der Festplatte wird anhand der Ausführungsdauer absoluter Lesezugriffe mit Hilfe von BIOS-Routinen bestimmt. Da Festplatten-Cache-Programme diesen Wert und den daraus errechneten Interleave-Faktor (siehe dazu DOS 10'88, Seite 126, DOS 9'89, Seite 30 und DOS Extra 9, Seite 72f) stark beeinflussen, ist eine objektive Abschätzung dieser Angaben nur ohne derartige Programme durchführbar. Die durchschnittliche Zugriffszeit kann infolge des angewandten Verfahrens von Fall zu Fall um bis zu 5 Millisekunden schwanken.

Die physikalischen Daten des Rechners werden anhand entsprechender BIOS- und MS-DOS-Routinen oder durch das Auswerten verschiedener BIOS-Variablen im Speichersegment 40hex ermittelt. Den Prozessortyp und -takt sowie die Anzahl der Waitstates bestimmt »config.pas« mit Hilfe des Zeitverhaltens verschiedener Maschinenbefehle.

Die Ausgabe der Daten läßt sich auch in eine Datei umlenken, weil »config.pas« zur Zeichenausgabe ausschließlich MS-DOS-Routinen verwendet. Die Unit »Crt« dürfen Sie allerdings nicht in das Programm einbinden. Die Ausgabeumleitung ließe sich dann nämlich nicht mehr durchführen, und die Werte für die Zeichenausgabe wären auch nicht mehr korrekt, weil Turbo Pascal in diesem Fall direkt in den Bildschirmspeicher schreibt.

Bei der Bestimmung der Ausstattung Ihres Rechners (hinsichtlich Quantität und Qualität) unterliegt »config.pas« jedoch auch einigen Einschränkungen:

- Bei 80386- und 80486-Prozessoren ermittelt das Programm keine Waitstates, weil das hier benutzte Verfahren nur auf 8- und 16-Bit-Rechnern richtige Werte liefert.
- Bei EGA- und VGA-Karten, bei denen ein 16-Bit-Bus festgestellt wurde, verzichtet »config.pas«

Computermodell	Prozessor	Geschwindigkeit (relativ zum IBM-PC)	Taktfrequenz (MHz)	Waitstates	Grafikkarte	Videodurchsatz direkt (Byte/s)	Videodurchsatz über BIOS (Byte/s)	Festplattenkapazität (MByte)	Übertragungsrate (KByte/s)	durchschnittliche Zugriffszeit (ms)	Zugriffszeit Spur-zu-Spur (ms)
PS/2 Model 80	80386	1167 %	20,0	-	VGA	220	28400	315	727	23	7
Kwem AT	80286	840 %	12,0	0	VGA	219	3800	32	284	25	7
Multitronic AT	80286	499 %	9,7	1	Hercules	182	3181	21	252	76	15
IBM AT/02	80286	300 %	6,0	1	EGA	100	1508	21	170	40	7
Commodore PC20	8088	207 %	9,5	0	Hercules	93	1074	21	122	77	17

Die Tabelle zeigt Beispiele für die mit »config.pas« ermittelten Daten

ebenfalls auf die Bestimmung von Waitstates.
- Die Prozessoren 80186 und 80188 werden nicht erkannt.
- Gameport-Adapter werden bei ATs nicht erkannt, weil die entsprechende BIOS-Variable bei ATs diese Karten nicht mehr berücksichtigt (obwohl im BIOS Joystick-Routinen implementiert sind).
- »config.pas« verändert den Vektor des Interrupt 50hex. Sollte es damit zu Problemen kommen, so müssen Sie einen anderen unbelegten Interrupt wählen.
- Sie dürfen das Programm »config.pas« nicht aus der integrierten Entwicklungsumgebung von Turbo Pascal starten, weil sonst bei den Festplattentests – aufgrund der Lage des Datenpuffers – für den Compiler wichtige Datenbereiche überschrieben werden.

(Michael Holin/zi)

Name	config.pas
Sprache	Turbo Pascal ab Version 4.0
Grafikkarte	jede
Besonderheiten	Programm NICHT aus der Pascal-Entwicklungsumgebung starten!

```
[145]    1: { Programmname: CONFIG.PAS   }
[V94]    2: { (C)  Everts & Hagedorn     }
[ZB4]    3: {      DOS International     }
[000]    4:
[3T3]    5: { Autor: Michael Holin
[000]    6:
[XB4]    7:   Das Programm NICHT aus der
[HD4]    8:   Entwicklungsumgebung von
[DX4]    9:   Turbo Pascal heraus starten,
[3A4]   10:   sondern als EXE-Datei auf
[AY4]   11:   eine Diskette oder Festplatte
[252]   12:   compilieren!
[U70]   13: }
[000]   14:
[4Z1]   15: {$I-,R-,D-}
[VG1]   16: uses dos;
[000]   17:
[NE4]   18: const ver='PC-Config V2.25  '+
[S04]   19:           '(C) DOS International  '+
[KX3]   20:           'Autor: Michael Holin';
[VX1]   21:       Int_Nr=$50;
[7A7]   22:  {Unbelegter Interrupt, der verbogen wird}
[KA1]   23:       timer=0;
[W35]   24:       zykl=838;                    {ns/Zyklus}
[S65]   25:       zyklinv=1193033;            {1/838.2 ns}
[0N4]   26:       x:array [0..1] of string[4]=
[A32]   27:         ('Nein','Ja  ');
[000]   28:
[000]   29:
[7H3]   30: var regs:registers;
[8L5]   31:     ax,buff,faktor,NullWert,PuffSeg,
[SA5]   32:     printB,printD,VideoWaits:word;
[BU2]   33:     i,drive:integer;
[MA4]   34:     kapa,sektoren,old,scroll,
[UF4]   35:     VideoDurchS,takt:longint;
[137]   36:     platte,kontrol,tport,tstart,tstop:byte;
[905]   37:     AT,EGA,VGA,cache,BUS16:boolean;
[875]   38:     testherc:  word absolute $B700:0;
[U85]   39:     ID:        byte absolute $F000:$FFFE;
[XH4]   40:     ram:       word absolute 0:$413;
[CW4]   41:     konfig:    word absolute 0:$410;
[B65]   42:     VideoMode: byte absolute 0:$449;
[6R4]   43:     zeit:      word absolute 0:$46C;
[N65]   44:     VideoInfo: byte absolute 0:$487;
[8K4]   45:     PS2:       word absolute 0:$40E;
[BV4]   46:     Kombi:     byte absolute 0:$48F;
[000]   47:
[CS3]   48: procedure StartTimer;
[MU0]   49: begin
[7K3]   50:  {Setzt Timer auf 65535
[UX4]   51:   Dieser wird nun alle 838 ns um
[3Z2]   52:   eins verringert}
[KU1]   53:   inline($FA);
[H34]   54:   port[kontrol]:=tstart;
[K33]   55:   port[tport]:=$FF;
[K33]   56:   port[tport]:=$FF;
[ZN0]   57: end;
[000]   58:
[H94]   59: function ReadTimer:word;
[MU0]   60: begin
[4Y3]   61:   port[kontrol]:=tstop;
[WT5]   62:   ReadTimer:=65535-port[tport] shl 8
[8F2]   63:              -port[tport];
[NU1]   64:   inline($FB);
[ZN0]   65: end;
[000]   66:
[PF4]   67: procedure TestVideoSpeed;
[VU5]   68:  {testet Geschw. des Bildschirms}
[Y02]   69: var i:word;
[GR4]   70:     source:longint absolute 0:0;
[UZ4]   71:     Mdest:longint absolute $B000:0;
[YZ4]   72:     Cdest:longint absolute $B800:0;
[HB3]   73:     NW,Screenbase:word;
[MU0]   74: begin
[T32]   75:   StartTimer;
[KE2]   76:   NW:=ReadTimer;
[NY3]   77:   if VideoMode=7 then begin
[4P5]   78:  {VideoRAM-Durchsatz bestimmen}
[5C3]   79:  {für Monochrom}
[T32]   80:       StartTimer;
[SN3]   81:       for i:=1 to 100 do begin
[YX4]   82:         Mdest:=source; Mdest:=source;
[YX4]   83:         Mdest:=source; Mdest:=source;
[MR6]   84:  {400 Langworte ins Video-RAM schreiben}
[F16]   85:  {--> 800 Zeichen inkl. Attribut       }
[ZN0]   86:       end;
[BL4]   87:       VideoDurchS:=ReadTimer-NW;
[2V2]   88:       Screenbase:=$B000;
[4M0]   89:       end
[WP1]   90:   else begin
[HH2]   91:  {für Farbe}
[T32]   92:       StartTimer;
[SN3]   93:       for i:=1 to 100 do begin
[0X4]   94:         Cdest:=source; Cdest:=source;
[0X4]   95:         Cdest:=source; Cdest:=source;
[ZN0]   96:       end;
[BL4]   97:       VideoDurchS:=ReadTimer-NW;
[2W2]   98:       Screenbase:=$B800;
[ZN0]   99:   end;
[MF6]  100:   VideoDurchS:=zyklinv div VideoDurchS
[9Y1]  101:                * 800 div 1024;
[5W1]  102:   regs.ah:=37;
[RV2]  103:   regs.al:=Int_Nr;
[UN5]  104:  {Interrupt auf Screen umbiegen}
[YF3]  105:   regs.ds:=Screenbase;
[UW1]  106:   regs.dx:=0;
[302]  107:   msdos(regs);
[NB6]  108:   mem[Screenbase:0]:=$CF;     { $CF = IRET}
[T32]  109:   StartTimer;
[P63]  110:   intr(Int_Nr,regs);
[VN3]  111:   nullwert:=ReadTimer;
[F04]  112:  {Zeit für IRET messen}
[D03]  113:   for i:=0 to 4000 do
[5Y3]  114:     mem[Screenbase:i]:=$90;
[1R4]  115:  {Screen mit NOPs füllen}
[A44]  116:   mem[Screenbase:4000]:=$CF;
[T32]  117:   starttimer;
[P63]  118:   intr(Int_Nr,regs);
[V83]  119:  {NOPs ausführen}
[AN5]  120:   VideoWaits:=readtimer-NullWert;
[T32]  121:   StartTimer;
[H95]  122:  {6 Zeichen über INT 16 ausgeben}
[275]  123:   inline($b8/$20/$0e/$cd/$10/$b8/$20/$0e/
[P24]  124:          $cd/$10/$b8/$20/$0e/$cd/$10/$b8/
[R24]  125:          $20/$0e/$cd/$10/$b8/$20/$0e/$cd/
[T23]  126:          $10/$b8/$20/$0e/$cd/$10);
[2M3]  127:   printB:=ReadTimer-NW;
[NC3]  128:   printB:=printB div 6;
[P15]  129:  {Anzahl der Zyklen (838.2 ns)}
[T32]  130:   StartTimer;
[6N4]  131:  {Zeichenausgabe über DOS}
[4V4]  132:   write('         ');          {9 Spaces}
[3L4]  133:   printD:=(ReadTimer-NW) div 9;
[C44]  134:   for i:=1 to 25 do writeln;
[T32]  135:   StartTimer;
[9T5]  136:  {Eine Zeile scrollen über INT 16}
[5U3]  137:   inline($b8/$0a/$0e/$cd/$10);
[EM3]  138:   scroll:=ReadTimer-NW;
[HU1]  139:   regs.ah:=0;
[CC3]  140:   regs.al:=videomode;
[A72]  141:   intr(16,regs);
[3Z3]  142:  {Bildschirm löschen}
[ZN0]  143: end;
[000]  144:
[A73]  145: procedure TestEMS;
[5P4]  146:  {Bestimmt Expanded Memory}
[VY1]  147: var i:byte;
[G52]  148:     ems:boolean;
[ST4]  149:     emsname,intname:string[8];
[MU0]  150: begin
[000]  151:
[ZK5]  152:  {Segment ermitteln, auf den der
[PG5]  153:   Interrupt-Vektor 67hex zeigt  }
[SX1]  154:   regs.ah:=$35;
[PY1]  155:   regs.al:=$67;
[302]  156:   msdos(regs);
[3C3]  157:   emsname:='EMMXXXX0';
[1B6]  158:  {Steht dieser String im Segment ES ab
[W35]  159:   Byte 10, dann ist der EMS-Manager
[2H4]  160:   installiert                       }
```

Listing. Mit »config.pas« erfahren Sie, was in Ihrem PC steckt

```
[RW1]  161:    intname:='';
[AG2]  162:    for i:=Ø to 7 do
[ØZ2]  163:      intname:=intname+
[1V3]  164:                 chr(mem[regs.es:1Ø+i]);
[KX3]  165:    ems:=(emsname=intname);
[VM2]  166:    if ems then begin
[YC4]  167:      write('Expanded Memory V');
[XX1]  168:      regs.ah:=$46;
[335]  169:   {Versionsnummer bestimmen}
[ZC2]  17Ø:      intr($67,regs);
[VB4]  171:      write(char(regs.al shr 4+48),
[6X3]  172:            '.',char(regs.al and 15+48),
[FB1]  173:            ' .....: ');
[TX1]  174:      regs.ah:=$42;
[ZC2]  175:      intr($67,regs);
[DE6]  176:   {Gesamtanzahl der Seiten bestimmen}
[K93]  177:      writeln(regs.dx*16,
[K15]  178:           ' KByte, davon sind noch frei: ',
[X43]  179:           regs.bx*16,' KByte');
[ZNØ]  18Ø:    end;
[ZNØ]  181: end;
[ØØØ]  182:
[KR3]  183: procedure TestEGABus;
[6E5]  184:  {EGA/VGA mit 8- oder 16-Bit-Bus?}
[4P3]  185: var i,loop,z1,z2:word;
[RW4]  186:     even: longint absolute $aØØØ:Ø;
[6U4]  187:     odd : longint absolute $aØØØ:1;
[MUØ]  188: begin
[683]  189:     loop:=faktor div 5;
[T32]  19Ø:     StartTimer;
[B54]  191:      for i:=1 to loop do begin
[6Z3]  192:        even:=even; even:=even;
[6Z3]  193:        even:=even; even:=even;
[ZNØ]  194:      end;
[SB2]  195:     z1:=ReadTimer;
[T32]  196:     StartTimer;
[B54]  197:      for i:=1 to loop do begin
[Y33]  198:        odd:=odd; odd:=odd;
[Y33]  199:        odd:=odd; odd:=odd;
[ZNØ]  2ØØ:      end;
[VB2]  2Ø1:     z2:=ReadTimer;
[FW5]  2Ø2:  {Differenz ? Ja: 16 Bit; Nein: 8 Bit}
[LH5]  2Ø3:     if z1+1Ø<z2 then write('16-Bit-Bus, ')
[VX3]  2Ø4:                 else write('8-Bit-Bus, ');
[8F4]  2Ø5:     if z1+1Ø<z2 then BUS16:=true;
[ZNØ]  2Ø6: end;
[ØØØ]  2Ø7:
[EY2]  2Ø8: procedure Search;
[5X5]  2Ø9:  {sucht nach BIOS-Erweiterungen}
[G23]  21Ø: var segment:word;
[X52]  211:     i,lo,hi:byte;
[ØN3]  212:     b:array [1..4] of byte;
[MUØ]  213: begin
[UE2]  214:   segment:=$CØØØ;
[1XØ]  215:   hi:=Ø;
[LA4]  216:   while segment<$e1ØØ do begin
[XW5]  217:     if memw[segment:Ø]=$AA55 then begin
[U83]  218:       lo:=segment and 255;
[T23]  219:       hi:=segment shr 8;
[VX4]  22Ø:       b[4]:=lo and 15; b[3]:=lo shr 4;
[BW4]  221:       b[2]:=hi and 15; b[1]:=hi shr 4;
[V15]  222:       for i:=1 to 4 do if b[i]<1Ø then
[9Ø3]  223:         write(chr(b[i]+48))
[LØ4]  224:       else write(chr(b[i]+55));
[Y12]  225:       write('hex ');
[ZNØ]  226:     end;
[CU2]  227:     inc(segment,$1ØØ);
[ZNØ]  228:   end;
[6U5]  229:   if hi=Ø then write(#13) else writeln;
[ZNØ]  23Ø: end;
[ØØØ]  231:
[1Y2]  232: procedure BIOSID;
[KZ5]  233:  {Sucht Copyright-Meldung im BIOS}
[C72]  234: var i,j:word;
[2N4]  235:     bios:array[Ø..32768] of byte
[533]  236:          absolute $FØØØ:$8ØØØ;
[MUØ]  237: begin
[WPØ]  238:   j:=Ø;
[DB4]  239:   if AT then i:=Ø else i:=16384;
[P31]  24Ø:   repeat
[M33]  241:     if (bios[i]=4Ø) and
[EZ3]  242:        (bios[i+2]=41) then begin
[R93]  243:  { »( )« suchen }
[P13]  244:        if (bios[i+1]=99) or
[UZ3]  245:           (bios[i+1]=67) then begin
[5R4]  246:  { »(c)« oder »(C)« ? }
[P31]  247:           repeat
[Z93]  248:             write(chr(bios[i]));
[951]  249:             inc(i);
[B51]  25Ø:             inc(j);
[GS4]  251:           until (bios[i]<14) or (j>5Ø) or
[DK2]  252:                 (bios[i]=46) or
[ØL2]  253:                 ((bios[i]=32) and
[DF2]  254:                 (bios[i+1]=32));
[N71]  255:           i:=32767;
[ZNØ]  256:         end;
[ZNØ]  257:       end;
[951]  258:       inc(i);
[N52]  259:     until i=32768;
[XF1]  26Ø:     writeln;
[ZNØ]  261: end;
[ØØØ]  262:
[VK5]  263: procedure Festpl(Funktion,Spur,
[TU3]  264:                  Sek,Anz,Messung:byte;
[8P2]  265:                  var Zeit:word);
[MUØ]  266: begin
[V43]  267:   regs.ah:=Funktion;
[YX5]  268:   regs.dl:=$8Ø+drive;            {Laufwerk}
[385]  269:   regs.dh:=Ø;                    {Kopfnr.}
[ØR5]  27Ø:   regs.ch:=Spur;                 {Spurnr.}
[ØV5]  271:   regs.cl:=Sek;                {Sektornr.}
[A86]  272:   regs.al:=Anz;          {Anzahl Sektoren}
[T27]  273:   regs.es:=PuffSeg;   {Segment des Puffers}
[1C6]  274:   regs.bx:=Ø;          {Offset des Puffers}
[TW4]  275:   if messung=1 then StartTimer;
[P72]  276:   intr(19,regs);
[8T2]  277:   if messung=1 then
[YG4]  278:     Zeit:=ReadTimer-nullwert;
[ZNØ]  279: end;
[ØØØ]  28Ø:
[SP3]  281: procedure InterLeave;
[357]  282:  {Bestimmt Datendurchsatz der Festplatte}
[9E3]  283: var i,sek,last:word;
[M94]  284:     lastd,durchs,max:longint;
[MUØ]  285: begin
[8A6]  286:   write('Maximaler Daten-Durchsatz..: ');
[5X2]  287:   Festpl(2,1Ø,1,1,Ø,i);
[UPØ]  288:   i:=Ø;
[4V3]  289:   for sek:=4 to 5Ø do begin
[RØ6]  29Ø:  {max. 5Ø Sektoren auf einmal lesen}
[MD2]  291:     lastd:=durchs;
[5C1]  292:     last:=i;
[2D3]  293:     Festpl(2,1Ø,1,sek,1,i);
[PR3]  294:     if regs.ah<>Ø then begin
[B85]  295:        writeln('kann nicht bestimmt '+
[9C4]  296:                'werden! Fehler: ',regs.ah);
[DUØ]  297:        halt;
[ZNØ]  298:     end;
[LP1]  299:     durchs:=1;
[7A3]  3ØØ:     max:=512*6Ø*sektoren;
[FT5]  3Ø1:  {maximaler Durchsatz in Byte/s}
[ZØ6]  3Ø2:     durchs:=durchs*zyklinv*512 div i*sek;
[V15]  3Ø3:     if ((i+2ØØØ<last) and not (sek<6)) or
[N13]  3Ø4:        (sek=5Ø) then begin
[2C3]  3Ø5:        write(lastd div 1Ø24,
[TK2]  3Ø6:              ' KByte/s  ==>  ');
[Ø22]  3Ø7:        sek:=5Ø; i:=Ø;
[P31]  3Ø8:        repeat
[951]  3Ø9:          inc(i);
[HZ2]  31Ø:          old:=(max div i)-15;
[ØØ3]  311:        until (lastd>old);
[D71]  312:        max:=Ø;
[2D2]  313:        if i<2 then i:=2;
[T95]  314:        writeln('Wahrsch. Interleave: ',
[WLØ]  315:                i-1);
[ZNØ]  316:     end;
[ZNØ]  317:   end;
[ZNØ]  318: end;
[ØØØ]  319:
[UT4]  32Ø: procedure Access(zyl:word);
[P47]  321:  {ermittelt Zugriffszeit der Festpatte}
[VT4]  322: var i,Spur,z1,z2,dummy:word;
[MUØ]  323: begin
[XP6]  324:   write('Durchschnittl. Zugriffszeit: ');
[1T1]  325:   randomize;
[C12]  326:   kapa:=zeit;
[AN3]  327:   for i:=Ø to 99 do begin
[4T2]  328:     write(99-i:2,#8,#8);
[K73]  329:     Spur:=random(zyl);
[KH5]  33Ø:     Festpl(12,Spur and 255,(Spur and 768)
[EX2]  331:            shr 2+1,1,Ø,dummy);
[PR3]  332:     if regs.ah<>Ø then begin
[B85]  333:        writeln('kann nicht bestimmt '+
[9C4]  334:                'werden! Fehler: ',regs.ah);
[DUØ]  335:        halt;
[ZNØ]  336:     end;
[ZNØ]  337:   end;
[GR2]  338:   kapa:=zeit-kapa;
[HØ4]  339:   write(kapa *1Ø div 18,' ms');
[Ø81]  34Ø:   kapa:=Ø;
[BP3]  341:   Festpl(12,1,1,1,Ø,dummy);
[K34]  342:   for Spur:=1 to 5Ø do begin
[8Ø5]  343:  {Bestimme Spur-zu-Spur-Zeit}
[AR3]  344:     Festpl(12,Spur,1,1,1,z1);
[L22]  345:     inc(kapa,z1);
[ZNØ]  346:   end;
[S85]  347:   kapa:=kapa div 5ØØØ*zykl div 1ØØØØ;
[Ø56]  348:   writeln(',  Spur zu Spur: ',kapa,' ms');
[LY2]  349:   Spur:=random(255);
```

```
[YE3]  35Ø:    for i:=1 to 2 do begin
[7K4]  351:   {Teste auf Cache-Programm}
[6A4]  352:      Festpl(2,Spur,1,1,1,dummy);
[251]  353:      z2:=z1;
[4X3]  354:      Festpl(2,Spur+1,1,1,1,z1);
[ZNØ]  355:    end;
[H95]  356:    if z2+7ØØ-z1>7ØØØ then cache:=true;
[ZNØ]  357: end;
[ØØØ]  358:
[ØØØ]  359:
[973]  36Ø: procedure TestCPU;
[K56]  361:  {ermittelt Prozessor, Geschw., Takt
[YØ3]  362:   und Waitstates}
[GØ4]  363: var i,j,loop,z1,z2:word;
[EE2]  364:     ergeb:longint;
[LR4]  365:     even:longint absolute $1ØØØ:Ø;
[1N4]  366:     odd :longint absolute $1ØØØ:1;
[MR4]  367:     garp:longint absolute $1ØØØ:3;
[EX4]  368:     bus,buszyklus,AAMtime:byte;
[7T3]  369:     NEC,I386,BUS32:boolean;
[MUØ]  37Ø: begin
[5W1]  371:   regs.ah:=37;
[RV2]  372:   regs.al:=Int_Nr;
[3P5]  373:  {Interrupt auf Puffer umbiegen}
[AZ2]  374:   regs.ds:=PuffSeg;
[UW1]  375:   regs.dx:=Ø;
[3Ø2]  376:   msdos(regs);
[L26]  377:   mem[PuffSeg:Ø]:=$CF;          { $CF = IRET}
[T32]  378:   StartTimer;
[P63]  379:   intr(Int_Nr,regs);
[VN3]  38Ø:   NullWert:=ReadTimer;
[FØ4]  381:  {Zeit für IRET messen}
[EX1]  382:   NEC:=false;
[5X1]  383:   I386:=false;
[V52]  384:   BUS32:=false;
[CE2]  385:   if AT then begin
[6K1]  386:     loop:=15Ø;
[C82]  387:     ergeb:=382ØØØØ;
[H92]  388:     buszyklus:=2;
[4MØ]  389:     end
[WP1]  39Ø:   else begin
[FD1]  391:     loop:=25;
[F22]  392:     ergeb:=667ØØØ;
[L92]  393:     buszyklus:=4;
[ZNØ]  394:   end;
[T32]  395:   StartTimer;
[LV2]  396:  {die hoffentlich
[555]  397:   repräsentative Zeitschleife}
[A54]  398:   for i:=Ø to loop do begin
[3G4]  399:     j:=mem[i:i]+(regs.ah shl 3)-
[TF2]  4ØØ:        succ(kapa) shr 4;
[N35]  4Ø1:     j:=pred(j)+ram and 3 or not even;
[ZNØ]  4Ø2:   end;
[E42]  4Ø3:   j:=ReadTimer;
[N83]  4Ø4:   faktor:=ergeb div j;
[YP3]  4Ø5:  {Ausgabe in Prozent}
[5Ø4]  4Ø6:   write(faktor,'%      CPU: ');
[M13]  4Ø7:   if not AT then begin
[383]  4Ø8:     loop:=faktor div 4;
[T32]  4Ø9:     StartTimer;
[B54]  41Ø:     for i:=1 to loop do begin
[6Z3]  411:       even:=even; even:=even;
[6Z3]  412:       even:=even; even:=even;
[ZNØ]  413:     end;
[SB2]  414:     z1:=ReadTimer;
[T32]  415:     StartTimer;
[B54]  416:     for i:=1 to loop do begin
[Y33]  417:       odd:=odd; odd:=odd;
[Y33]  418:       odd:=odd; odd:=odd;
[ZNØ]  419:     end;
[VB2]  42Ø:     z2:=ReadTimer;
[384]  421:  {8-Bit-XT oder 16-Bit-XT?}
[RH3]  422:     if z1+1Ø<z2 then bus:=16
[KZ1]  423:                 else bus:=8;
[983]  424:     loop:=faktor div 1Ø;
[A71]  425:     j:=23456;
[T32]  426:     StartTimer;
[R93]  427:     for i:=1 to loop do
[Ø54]  428:       ergeb:=j div i div i div i;
[SB2]  429:     z1:=ReadTimer;
[2WØ]  43Ø:     j:=72;
[T32]  431:     StartTimer;
[3R5]  432:     for i:=1 to loop do ergeb:=j*i*i*i;
[VB2]  433:     z2:=ReadTimer;
[7S6]  434:  {Division gegenüber Multiplikation}
[K12]  435:     if bus=8 then
[ØD4]  436:       if z1>z2+5Ø then write('8Ø88')
[WP1]  437:       else begin
[ØB2]  438:         write('NEC V2Ø');
[ZR1]  439:         nec:=true;
[ZNØ]  44Ø:       end;
[662]  441:     if bus=16 then
[YC4]  442:       if z1>z2+5Ø then write('8Ø86')
[WP1]  443:       else begin
[2B2]  444:         write('NEC V3Ø');
[ZR1]  445:         nec:=true;
[ZNØ]  446:       end;
[4MØ]  447:    end
[WP1]  448:    else begin
[531]  449:  { AT }
[383]  45Ø:     loop:=faktor div 4;
[T32]  451:     StartTimer;
[A54]  452:     for i:=Ø to loop do begin
[Y33]  453:       odd:=odd; odd:=odd;
[Y33]  454:       odd:=odd; odd:=odd;
[ZNØ]  455:     end;
[SB2]  456:     z1:=ReadTimer;
[T32]  457:     StartTimer;
[A54]  458:     for i:=Ø to loop do begin
[2Ø4]  459:       garp:=garp; garp:=garp;
[2Ø4]  46Ø:       garp:=garp; garp:=garp;
[ZNØ]  461:     end;
[VB2]  462:     z2:=ReadTimer;
[GA3]  463:  {16- oder 32-Bit-AT?}
[A33]  464:     if z1+9<z2 then begin
[LV1]  465:       I386:=true;
[E22]  466:       BUS32:=true;
[P52]  467:       write('8Ø386');
[4MØ]  468:     end
[WP1]  469:     else begin
[K93]  47Ø:  {8Ø286 oder 8Ø386SX?}
[T32]  471:       StartTimer;
[2D3]  472:       for i:=Ø to faktor do
[MU3]  473:         j:=((i shl 8) shr 8) shl 8;
[444]  474:       z1:=ReadTimer-NullWert;
[T32]  475:       StartTimer;
[M65]  476:  {Beim 386 ist die Zeit für Shift-
[U76]  477:   Operationen von Bits unabhänig von
[WW4]  478:   der Anzahl der Bits                }
[2D3]  479:       for i:=Ø to faktor do
[4T3]  48Ø:         j:=((i shl 2) shr 2) shl 2;
[744]  481:       z2:=ReadTimer-NullWert;
[1K4]  482:       if z1>z2+4Ø then write('8Ø286')
[WP1]  483:       else begin
[AF2]  484:         write('8Ø386SX');
[LV1]  485:         I386:=true;
[ZNØ]  486:       end;
[ZNØ]  487:     end;
[ZNØ]  488:   end;
[KH5]  489:   if AT then    if I386 then AAMtime:=17
[AR2]  49Ø:                          else AAMtime:=16;
[RS5]  491:   if not AT then if NEC then AAMtime:=15
[3S2]  492:                          else AAMtime:=77;
[BØ3]  493:   for i:=Ø to 3ØØØ do
[SN4]  494:     memw[PuffSeg:i shl 1]:=$ØAD4;
[AB5]  495:  {Puffer mit D4ØAhex füllen}
[UØ7]  496:   mem[PuffSeg:i*2+2]:=$CF;  {IRET ans Ende}
[T32]  497:   StartTimer;
[P63]  498:   intr(Int_Nr,regs);
[644]  499:  {Die 3ØØ1 AAMs ausführen}
[444]  5ØØ:   z1:=ReadTimer-NullWert;
[ØU4]  5Ø1:  {Reine Laufzeit bestimmen}
[N81]  5Ø2:   takt:=1;
[4P3]  5Ø3:   takt:=AAMtime*1ØØØØØ div
[PM3]  5Ø4:         (takt*z1*zykl div 3ØØ1);
[ØU1]  5Ø5:  {Takt * 1ØØ}
[BY3]  5Ø6:   write(', ',takt div 1ØØ,'.',
[8Y2]  5Ø7:         takt mod 1ØØ,' MHz');
[Ø43]  5Ø8:   if bus=8 then begin
[E74]  5Ø9:  {Bestimmung von WAITs}
[BØ3]  51Ø:     for i:=Ø to 3ØØØ do
[7L4]  511:       memw[PuffSeg:i shl 1]:=$9Ø9Ø;
[HV3]  512:     mem[puffseg:6ØØ1]:=$CF;
[T32]  513:     starttimer;
[ØL3]  514:  {6ØØØ NOPs ausführen}
[P63]  515:     intr(Int_Nr,regs);
[444]  516:     z1:=readtimer-nullwert;
[481]  517:     kapa:=1;
[FK5]  518:     kapa:=kapa*z1*zykl div 6ØØ1*takt div
[S41]  519:           1ØØØØØ -4;
[1E5]  52Ø:     writeln(', ',kapa,' Waitstate(s)');
[4MØ]  521:   end
[WP1]  522:   else begin
[DD1]  523:  {16 Bit}
[XF6]  524:  {MOV AX,ØØØØ im Vergleich zu MOV AX,ØØØ3
[NW5]  525:   Anhand der Differenz lassen sich die
[K35]  526:   Waitstates erkennen                  }
[5W3]  527:     for i:=Ø to 6ØØØ do begin
[8P3]  528:       mem[puffseg:i*3]  :=$A1;
[ST3]  529:       mem[puffseg:i*3+1]:=$ØØ;
[UT3]  53Ø:       mem[puffseg:i*3+2]:=$ØØ;
[ZNØ]  531:     end;
[DY3]  532:     mem[puffseg:i*3+3]:=$CF;
[T32]  533:     starttimer;
[P63]  534:     intr(Int_Nr,regs);
[444]  535:     z1:=readtimer-nullwert;
[TN3]  536:  {Zeit für MOV AX,ØØØØ}
[HØ3]  537:     for i:=Ø to 6ØØØ do
[4U3]  538:       mem[puffseg:i*3+1]:=$Ø3;
[T32]  539:     starttimer;
[P63]  54Ø:     intr(Int_Nr,regs);
[744]  541:     z2:=readtimer-nullwert;
[ZN3]  542:  {Zeit für MOV AX,ØØØ3}
[481]  543:     kapa:=1;
[U15]  544:     kapa:=kapa*(z2-z1)*zykl*takt div
```

Listing. Mit »config.pas« erfahren Sie, was in Ihrem PC steckt

```
[L44]  545:            6ØØ1 div 1ØØØØØ -buszyklus;
[WC3]  546:      if BUS32 then writeln
[C56]  547:      else writeln(', ',kapa,' Waitstate(s)')
       ;
[ZNØ]  548:    end;
[BB6]  549:   {Keine Waitstates-Bestimmung bei 8Ø386
[D86]  55Ø:    (Da wird sich sowieso keiner einig!) }
[ZNØ]  551: end;
[ØØØ]  552:
[4Z2]  553:  {===== MAIN =====}
[ØØØ]  554:
[MUØ]  555: begin
[162]  556:    BUS16:=false;
[W32]  557:    cache:=false;
[RA4]  558:   {Kein Cache installiert}
[UH3]  559:    PuffSeg:=(ram-3ØØØØ)*64;
[7P4]  56Ø:   {Ca. 28 KByte vor RAM-Ende}
[932]  561:    kontrol:=$43;
[LV2]  562:    tport :=$4Ø+timer;
[YD6]  563:   {Startwerte für Kurzzeitmessung}
[483]  564:    tstop :=timer shl 6;
[433]  565:    tstart:=$34+tstop;
[KC2]  566:    if timer=2 then
[K74]  567:      port[$61]:=port[$61] or 1;
[HU2]  568:    TestVideoSpeed;
[XD5]  569:   {Werte für Videogeschw. holen}
[CA2]  57Ø:    writeln(ver);
[XF1]  571:    writeln;
[BZ5]  572:    write('Dieser Rechner ist ein.....: ');
[GN1]  573:    AT:=false;
[5N5]  574:    if (ID=252) or (ID=248) then AT:=true;
[NM1]  575:    case ID of
[GA2]  576:      $ff:write('PC');
[D53]  577:      $fe:if PS2=$9FCØ then
[CU3]  578:            writeln('PS/2 Model 3Ø')
[KR3]  579:          else writeln('PC/XT');
[553]  58Ø:      $fc:if PS2=$9FCØ then
[P74]  581:            writeln('PS/2 Model 5Ø/6Ø')
[W43]  582:          else writeln('AT');
[T33]  583:      $f8:if PS2=$9FCØ then
[W74]  584:            writeln('PS/2 Model 7Ø/8Ø')
[BM3]  585:          else writeln('AT 386');
[WP1]  586:      else begin
[Y13]  587:        writeln; writeln;
[YY5]  588:        writeln('Der Typ dieses Rechners '+
[MH4]  589:                'kann nicht identifiziert '+
[EW3]  59Ø:                'werden. Im folgenden');
[4Y5]  591:        writeln('wird davon ausgegangen, '+
[E24]  592:                'daß es sich um ein AT-'+
[373]  593:                'Modell handelt.');
[RE5]  594:        writeln('Bitte melden Sie diese '+
[NU3]  595:                'Zahl und das Computer'+
[574]  596:                'modell an den Autor: ',ID);
[XF1]  597:        writeln;
[GK1]  598:        AT:=true;
[ZNØ]  599:      end;
[ZNØ]  6ØØ:    end;
[943]  6Ø1:    write('BIOS vom ');
[AG2]  6Ø2:    for i:=Ø to 7 do
[FS4]  6Ø3:      write(chr(mem[$fØØØ:$fff5+i]));
[MT2]  6Ø4:    write('.....: ');
[S91]  6Ø5:    BIOSID;
[TE6]  6Ø6:    write('Relative Geschw. zum IBM-PC: ');
[WG1]  6Ø7:    TestCPU;
[N86]  6Ø8:    write('BIOS-Erweit. gefunden bei..: ');
[N81]  6Ø9:    Search;
[9K6]  61Ø:    write('Arbeitsspeichergröße.......: ',
[F62]  611:          ram,' KByte');
[1Z4]  612:    writeln(', davon sind belegt: ',
[B74]  613:            prefixseg div 64,' KByte');
[CE2]  614:    if AT then begin
[ØZ1]  615:      regs.ah:=136;
[T62]  616:      intr(21,regs);
[DØ6]  617:      writeln('Extended Memory.....: '
       ,
[ZS3]  618:              regs.ax,' KByte frei');
[ZNØ]  619:    end;
[FG1]  62Ø:    TestEMS;
[PV3]  621:    VGA:=false; EGA:=false;
[8T2]  622:   {Teste auf VGA}
[3B2]  623:    regs.ax:=$1BØØ;
[DZ2]  624:    regs.es:=PuffSeg;
[MW1]  625:    regs.bx:=Ø;
[A72]  626:    intr(16,regs);
[NG6]  627:    write('Installierte Grafikkarte...: ');
[WM4]  628:    if regs.al=$1B then VGA:=true
[WP1]  629:    else begin
[RS2]  63Ø:   {Teste auf EGA}
[3W1]  631:      regs.ah:=18;
[EW1]  632:      regs.bl:=16;
[RW1]  633:      regs.cx:=Ø;
[A72]  634:      intr(16,regs);
[7H4]  635:      if regs.cx>Ø then EGA:=true;
[ZNØ]  636:    end;
[UA4]  637:    if EGA then write('EGA mit ');
[WB4]  638:    if VGA then write('VGA mit ');
[2Z5]  639:    if AT and (EGA or VGA) then TestEGABus;
[DY3]  64Ø:    if (EGA or VGA) then begin
```

```
[ØD4]  641:      case (VideoInfo and 112) of
[RS2]  642:        112,16: write('512');
[Z32]  643:             Ø: write('64');
[P92]  644:            32: write('128');
[Y92]  645:            64: write('192');
[3M2]  646:          96,48: write('256');
[ZNØ]  647:      end;
[X43]  648:      write(' KByte und ');
[344]  649:      if (VideoInfo and 2)=2 then
[6Z2]  65Ø:        write('Monochrom')
[153]  651:      else write('Farb');
[PP2]  652:      write('monitor');
[ZNØ]  653:    end;
[SK3]  654:    if not (EGA or VGA) then
[NY3]  655:      if VideoMode=7 then begin
[LH2]  656:        testherc:=12345;
[AF3]  657:        if testherc=12345 then
[FS2]  658:          write('Hercules')
[7X2]  659:        else write('MDA');
[4MØ]  66Ø:      end
[WW2]  661:      else write('CGA');
[XF1]  662:    writeln;
[WT5]  663:    write('VideoRAM-Durchsatz.....: ',
[9Y3]  664:          VideoDurchS,' KByte/s');
[UT5]  665:    scroll:=scroll div 1ØØ*zykl div 1ØØØ;
[EY4]  666:    write(', Scroll: ',scroll div 1Ø,
[KB3]  667:          ',',scroll mod 1Ø,' ms');
[8L2]  668:    if not BUS16 then
[K44]  669:       write(', ',VideoWaits*takt
[5X3]  67Ø:             div 4ØØØ *zykl div 1ØØØØØ -4,
[EF2]  671:             ' VideoWaits');
[MA2]  672:    writeln(#13#1Ø,
[ØY4]  673:            'Zeichenausgabe BIOS, DOS...: ',
[585]  674:            zyklinv div printB,' Byte/s, ',
[455]  675:            zyklinv div printD,' Byte/s');
[P86]  676:    write  ('Momentaner Videomodus......: ',
[BA2]  677:            VideoMode,', ');
[GT2]  678:    case VideoMode of
[WP4]  679:      Ø: writeln('4Ø*25 Zeichen, S/W');
[Y65]  68Ø:      1: writeln('4Ø*25 Zeichen, farbig');
[8R4]  681:      2: writeln('8Ø*25 Zeichen, S/W');
[A75]  682:      3: writeln('8Ø*25 Zeichen, farbig');
[3A5]  683:      4: writeln('32Ø*2ØØ Punkte Grafik, '+
[5L1]  684:                 '4 Farben');
[5A5]  685:      5: writeln('32Ø*2ØØ Punkte Grafik, '+
[FØ1]  686:                 'S/W');
[LA5]  687:      6: writeln('64Ø*2ØØ Punkte Grafik, '+
[ZK1]  688:                 '2 Farben');
[FH4]  689:      7: writeln('8Ø*25 Zeichen im '+
[473]  69Ø:                'Monochrom-Modus');
[ZNØ]  691:    end;
[PØ4]  692:    regs.ah:=48; msdos(regs);
[CT5]  693:    writeln('DOS-Version..............: ',
[X53]  694:            regs.al,'.',regs.ah);
[1Z6]  695:    writeln('Mathematischer Coprozessor.: ',
[DV3]  696:            x[(konfig shr 1) and 1]);
[152]  697:    if not AT then
[ZA6]  698:    writeln('Gameadapter vorhanden......: ',
[1T3]  699:            x[(konfig shr 12) and 1]);
[3S6]  7ØØ:    writeln('Anzahl Diskettenlaufwerke..: ',
[FE3]  7Ø1:            (konfig shr 6) and 3+1);
[RE6]  7Ø2:    writeln('Serielle, parallele Ports..: ',
[ØM3]  7Ø3:            (konfig shr 9) and 7,', ',
[FL3]  7Ø4:            (konfig shr 14) and 3);
[UE2]  7Ø5:    i:=-1; drive:=-1;
[NU3]  7Ø6:    if AT and (PS2<>$9FCØ) then
[BR6]  7Ø7:    writeln('Kombi-Controller vorhanden.: ',
[SP2]  7Ø8:            x[(kombi and 1)]);
[P31]  7Ø9:    repeat
[951]  71Ø:      inc(i);
[9V1]  711:      regs.ah:=8;
[7C2]  712:      regs.dl:=$8Ø+i;
[P72]  713:      intr(19,regs);
[SS3]  714:      if regs.ah=Ø then begin
[6N4]  715:         inc(drive); platte:=regs.dl;
[144]  716:         ax:=(regs.cl and 192) shl 2;
[SY3]  717:         sektoren:=regs.cl and 63;
[MU4]  718:         kapa:=(regs.ch+ax+1)*sektoren;
[XP4]  719:         kapa:=kapa*(regs.dh+1) div 195;
[U51]  72Ø:    {1Ø/512}
[P65]  721:         write('Daten für Festplatte ',
[4Ø4]  722:               drive+1,' ....: ',kapa div 1Ø,
[ØW3]  723:               ',',kapa mod 1Ø,' MByte, ');
[T54]  724:         write(regs.dh+1,' Köpfe, ',
[764]  725:               regs.ch+ax+1,' Zylinder, ');
[S35]  726:         writeln(sektoren,' Sekt/Zyl');
[N22]  727:         InterLeave;
[573]  728:         Access(regs.ch+ax+1);
[DW4]  729:         if (regs.dl=1) or (platte=1) then
[WPØ]  73Ø:           i:=2;
[ZNØ]  731:      end;
[XR1]  732:    until (i=2);
[MØ2]  733:    if cache then
[K16]  734:      writeln('Ein Festplatten-Cache ist '+
[X55]  735:              'installiert! Interleave und '+
[ZE4]  736:              'Durchsatz stimmen nicht!');
[KNØ]  737: end.
```

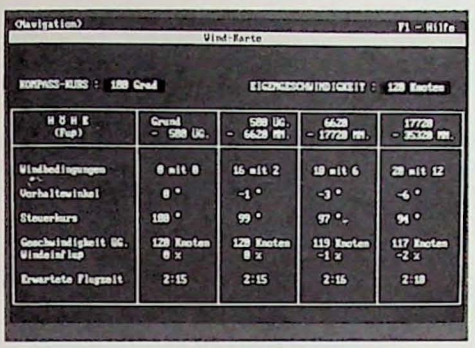
Listing. Mit »config.pas« erfahren Sie, was in Ihrem PC steckt

DOS SOFTWARE TOOLS

JETZT NEU Navigator

Navigator für den Microsoft Flugsimulator III

Flugsimulation naturgetreu und realistisch



Der "Navigator" wertet den Microsoft Flugsimulator zu einem äußerst realistischen und wirklichkeitsnahen Flugtrainer auf.
Ob anspruchsvolle Unterhaltung oder naturgetreues Flugtraining – mit dem "Navigator" können Sie alle Möglichkeiten des Microsoft Flugsimulators voll erschließen.

Die Leistungsmerkmale auf einen Blick:

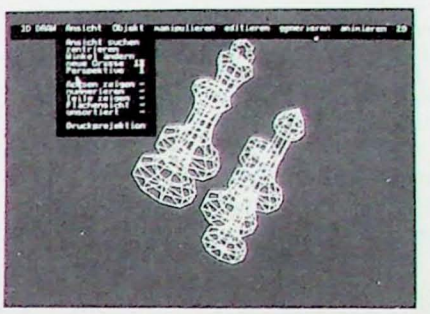
- ★ Genaue Daten von mehr als 400 verschiedenen Flughäfen
- ★ Alle Landschaftsdisketten von Sublogic verwendbar
- ★ Jeder Flughafen gibt realistische, das heißt wechselnde Wetterberichte aus
- ★ Tabellen und Grafiken informieren über Windeinfluß auf Kurs und Geschwindigkeit
- ★ Berechnung der vorgeschriebenen Reiseflughöhe für VFR- und IFR-Flüge
- ★ Präzise Landeanflugkarte für jeden angeflogenen Flughafen
- ★ Ausgabe eines kompletten, wirklichkeitsgetreuen Flugplans
- ★ Wetterbericht, Flugplan sowie alle Tabellen und Grafiken können ausgedruckt werden
- ★ Konfiguration nach eigenem Können und Geschmack
- ★ Ausführliches, deutsches Handbuch

Systemvoraussetzungen:
PC, XT, AT mit beliebiger Grafikkarte

nur 69,– DM*

3D-Draw Das komfortable CAD-Programm

Mit Lichtquellensimulation und Animation



★ Ansicht durch Eingabe der Projektionswinkel frei wählbar ★ Auf Wunsch lassen sich drei verschiedene 2D-Ansichten erzeugen ★ 2D-Editor zur schnellen Konstruktion von Objekten ★ Integrierter Taschenrechner für Konstruktionsdaten ★ Einstellbare Größe und perspektivische Verzerrung ★ Kombination von getrennt konstruierten Teilen ★ 3D-Funktion als Manipulationswerkzeug ★ Punkte, Kanten und Flächen sind frei editierbar ★ Drei Raumachsen mit 1600-Ebenen-Editor ★ Verdeckte Flächen ★ Lichtquellensimulator ★ Animationsroutine mit Größen- und Perspektivenveränderung von beliebiger Länge ★ Animationslinker ★ Verformen der Objekte durch Zufallszahl oder Verknüpfung mit mathematischen Formeln ★ Erzeugung von Rotations- und Translationskörpern und modifizierten Duplikaten ★ Programmierer-Schnittstelle, Objekte lassen sich mit einem Texteditor generieren ★ Vielzahl von Ausgabeformaten: 3D-Draw, Autocad-Script, Postscript, EPS-Format und HPGL-Plot-Format

Systemvoraussetzungen:
Computer: PC, XT, AT oder Kompatibler mit mindestens 640 KByte Arbeitsspeicher, ein oder zwei Diskettenlaufwerke beziehungsweise ein Diskettenlaufwerk und eine Festplatte
Grafikkarten: Hercules-, CGA-, EGA- oder VGA-Karte
Ausgabemedien: Epson-kompatibler Drucker, Ausgabe von Postscript-Dateien in Verbindung mit Postscript-Emulator, Postscript-fähiger Laserdrucker, Plotter mit HPGL-Standard
Sonstiges: Microsoft-kompatible Maus unbedingt erforderlich, MS-DOS ab Version 3.0

Jetzt Demonstrationsdiskette erhältlich!

nur 99,– DM*

GW-Basic-Toolboxen

Professionelle Leistung und komfortable Bedienung: kompatibel zum GW-Basic-Interpreter, gleichzeitiger Gebrauch mehrerer GW-Basic-Toolboxen, leistungsfähige Beispielprogramme inklusive.

GW-BASIC-TOOLBOX "Oberfläche"

Für komfortable Benutzeroberflächen mit Pulldown-Menüs und kompletter Mausunterstützung:
- ★ Vollständige Maussteuerung im Text- und Grafikmodus
- ★ Spezielle Menüleisten für häufig benötigte Eingaben
- ★ Popup-Menüs mit Meldungen bis zu fünf Zeilen Länge
- ★ Position, Farben und Rahmen aller Menüs frei wählbar
- ★ Unterstützung aller Bildschirmmodi von GW-Basic
- ★ Inklusive einem fertigen Malprogramm als Beispiel

GW-BASIC-TOOLBOX "DOS-Funktionen"

Mehr als 30 Funktionen für das MS-DOS-System:
- ★ Kompatibel zu MS-DOS von Version 2.1 bis 4.01
- ★ Laufwerke wechseln, aktuelles Laufwerk ermitteln etc.
- ★ Suchen nach Dateien mit den Wildcards "*" und "?"
- ★ Schnelle Bildschirmausgabe über Systemfunktionen
- ★ Zugang zu wichtigen, internen Daten von MS-DOS
- ★ Nutzung des Druckprogramms Print von GW-Basic aus
- ★ Inklusive Beispielprogramm: eine komplette DOS-Shell

GW-BASIC-TOOLBOX "Hercules-Grafik"

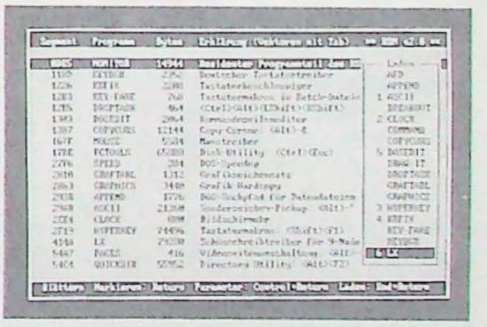
Programmieren der Hercules-Karte im Grafikmodus:
- ★ Routinen zum Ein- und Ausschalten des Grafikmodus
- ★ Zwei verfügbare Grafikseiten mit Umschaltmöglichkeit
- ★ Funktionen für Linie, Bogen, Kreis, Ellipse und Vieleck
- ★ Ausfüllen von Flächen mit beliebigen Punktmustern
- ★ Speichern und Restaurieren des Grafikbildschirms
- ★ Setzen, Löschen und Abfragen von einzelnen Punkten
- ★ Emulation der GW-Basic-Befehle "View" und "Window"
- ★ Inklusive Beispielprogramm: Präsentationsgrafik

Jede GW-Basic-Toolbox

nur DM 69,–*

Resident-Software-Manager

Menügesteuertes Löschen und Nachladen von residenten TSR-Programmen löst Ihre Speicherplatzprobleme: RSM



★ zeigt den noch freien Arbeitsspeicher ★ registriert geladene residente Programme ★ erinnert an die Aktivierungstastenkombination ★ informiert über benutzte Interrupt-Vektoren ★ löscht beliebige, speicherresidente Programme ★ lädt mehrere TSR-Programme in einem Arbeitsgang ★ bietet umfassende, komfortable Menüsteuerung ★ besitzt einen universellen Maustreiber ★ belegt lediglich 15 kByte Speicherplatz ★ informiert mit einem umfangreichen Handbuch

Für alle PCs, XTs, ATs und Kompatible

nur 69,– DM*

PC-Utilities

nur 69,– DM*

Softreset: Spannen Sie Ihre Nerven nicht weiterhin auf die Folter! Mit Softreset bleibt der komplette Speicherinhalt auch nach einem Reset erhalten und kann auf Diskette oder Festplatte gespeichert werden. **Zeitlupe PC:** Verlangsamen Sie Ihren Computer bis zu einem Bruchteil der Taktfrequenz. Mit Boot-Option! **Extras:** Der universell einsetzbare AT-Tastaturtreiber »T6« mit Quellcode.

MS-DOS-Utilities

nur 49,– DM*

Call: verschachtelt Aufrufe von Batch-Dateien. **Code:** Paßwortverschlüsselung. **Makeroom:** löscht Dateien gemäß Suffix. **Setdir, Restdir:** aktivieren nach Programmaufruf das Original-Verzeichnis. **Ty:** Hex-Dump, Binärformat oder Zeilennummer auf dem Drucker oder Bildschirm aus. **Extras:** »SCopy«, das professionelle Kopierprogramm, »Filed«, der Programmeditor.

* Sämtliche Programme sind wahlweise auf 5 1/4-Zoll- oder 3 1/2-Zoll-Diskette erhältlich. (Bitte bei Bestellung unbedingt angeben.) Unabhängig von der Anzahl der bestellten Artikel berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung. Unverbindliche Preisempfehlung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Abteilung DOS-Software · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Das Drucker-Lexikon

Die Sprache in Druckerhandbüchern ist häufig für Einsteiger ziemlich unverständlich. Wir haben nun die Grundbegriffe, die beim Umgang mit Druckern auftreten, in einem Lexikon zusammengefaßt und erklärt.

ASCII
»ASCII« ist die Abkürzung für »American Standard Code for Information Interchange«. Jedem Zeichen wird ein Code zugeordnet, der sich durch eine siebenstellige Binärzahl ausdrücken läßt. Auf diese Weise lassen sich hier 128 Zeichen darstellen. Da diese 128 Zeichen meist nicht ausreichen, wurde der sogenannte »erweiterte ASCII-Code« eingeführt. Dieser benutzt acht Bit und kann somit 256 Zeichen darstellen.

Baud
»Baud« ist die Maßeinheit, mit der die Geschwindigkeit beim seriellen Datenaustausch gemessen wird. Obwohl diese Einheit nicht die Geschwindigkeit mißt, hat es sich eingebürgert, für »Bit pro Sekunde« den Begriff »Baud« zu verwenden.

Bidirektional
Beim »bidirektionalen« Modus druckt der Druckkopf im Gegensatz zum unidirektionalen in beide Richtungen, also die erste Zeile von links nach rechts, die zweite von rechts nach links und so weiter. Dieses Verfahren erhöht die Druckgeschwindigkeit.

Blocksatz
Im »Blocksatz« werden Texte rechts- und linksbündig formatiert, so daß jeder Absatz wie ein Block wirkt. Diese Formatiermethode vermittelt ein wesentlich »geschlosseneres« Druckbild, als der Flattersatz.

Carriage Return
»Carriage Return«, oder abgekürzt »CR«, heißt auf Deutsch »Wagenrücklauf«. Dieser Befehl wird durch das Zeichen mit dem ASCII-Code 13 ausgelöst und bedeutet, daß der Druckkopf an den linken Rand der Zeile fahren soll.

Centronics-Schnittstelle
Die »Centronics-Schnittstelle« wird als »parallele Schnittstelle« bezeichnet. Sie ist eine normierte Verbindung zwischen Rechner und Drucker. Über acht Datenleitungen werden die Informationen byteweise übertragen.

cpi
Der Begriff »cpi« kommt aus dem Englischen und steht für »characters per inch«, also »Zeichen pro Zoll«. Diese Maßeinheit wird für die Druckbreite verwendet. Je mehr cpi eine Schrift hat, desto schmaler sind die einzelnen Zeichen.

cps
»cps« ist ebenfalls ein englischer Begriff und kürzt »characters per second« ab. Damit wird die Druckgeschwindigkeit gemessen.

Dip-Schalter
Aufgeschlüsselt steht »Dip« für »Dual Inline Package«, was soviel wie »Zweiwegeschaltereinheit« heißt. Genervte Benutzer nennen diese Steuerung auch »Mäuseklavier«. Mit Dip-Schaltern wird die Grundeinstellung des Druckers festgelegt. Viele moderne Drucker verzichten mittlerweile auf diese Art der Grundeinstellung.

Download
»Download« wird auch »benutzerdefinierter Zeichensatz« genannt. Der Anwender definiert hierbei das Aussehen und die Form der Zeichen selbst. Diese werden dann im Arbeitsspeicher des Druckers abgelegt und treten beim Ausdruck an die Stelle der Standardzeichen.

dpi
»dpi« ist eine englische Abkürzung und bedeutet »dots per inch«, also »Punkte pro Zoll«. dpi ist die Maßeinheit der Grafikauflösung bei Druckern. Der Standard bei Laserdruckern beträgt 300x300 dpi, bei 24-Nadel-Druckern 180x360 dpi.

Draft
»Draft« ist eine Schriftqualität, die für Kontrollausdrucke oder Listings bestimmt ist, da sie zwar schnell, aber nicht sehr schön ist.

Druckertreiber
Der Begriff »Druckertreiber« bezeichnet den Teil eines Anwendungsprogramms, das den Drucker anspricht.

Elite
Mit »Elite« wird eine Schrift mit der Zeichendichte von 12 dpi bezeichnet.

Emulation
Unter »Emulation« wird die Nachbildung des Verhaltens eines Druckers auf einem anderen verstanden. Viele Drucker emulieren beispielsweise den IBM-Proprinter oder den Epson LQ, während bei Laserdruckern die HP-Laserjet-Emulation Standard ist.

ESC/P
Der Begriff »ESC« ist die englische Abkürzung für »Epson Standard Code for Printers«. Die ESC/P-Sequenzen sind normierte Befehle für Druckersteuerungen. Diese Sequenzen sind auch unter dem Namen »Epson-Emulation« bekannt.

FF
»FF« ist die Abkürzung für »Formfeed« und bedeutet im Deutschen »Seitenvorschub«. Mit diesem Befehl wird das eingelegte Papier zum Beginn der nächsten Seite transportiert.

Flattersatz
Ein Text im »Flattersatz« ist ein linksbündig formatierter Text mit ungleich langen Zeilen, so daß der rechte Rand nicht bündig abschließt.

Font
»Font« ist das englische Wort für »Zeichensatz«. Es bezeichnet die Menge der druckbaren Zeichen einer Schriftart. Fast alle Drucker verfügen in der Standardausrüstung nur über eine eingeschränkte Anzahl von Schriftarten, so daß bei Bedarf sogenannte »Fontcards«, also »Schriftartkarten«, nachgekauft werden müssen.

Initialisieren
Der Drucker wird durch die »Initialisierung« in den Zustand zurückversetzt, den er nach dem Einschalten hatte. Allerdings bleibt der Inhalt des Druckpuffers erhalten.

Interface
»Interface« bedeutet im Deutschen »Schnittstelle«. Bei Druckern weist es – je nach Druckertyp – auf die parallele oder serielle Schnittstelle hin, durch die der Rechner mit dem Drucker verbunden ist.

LF
Abkürzung für »Linefeed«; dieser Befehl bewirkt den Weitertransport des Papiers um eine Zeile.

LQ
»LQ« kommt aus dem Englischen und bedeutet »Letter Quality«, also »Briefqualität«. In dieser Schriftart sind die Buchstaben kaum mehr von Buchstaben, die mit Typenrad- oder Laserdruckern erzeugt wurden, zu unterscheiden.

NLQ
Der Bezeichnung »NLQ« kommt aus dem Englischen und bedeutet »Near Letter Quality«, was mit »Fast-Brief-Qualität« übersetzt werden könnte. Das heißt, die Schrift ist fast so gut wie LQ. Meist wird die Schönschrift bei 9-Nadel-Druckern als »NLQ« bezeichnet, weil sie an die LQ-Qualität der 24-Nadel-Drucker nicht ganz heranreicht.

Papierende-Sensor
Viele Drucker sind mit »Papierende-Sensoren« ausgestattet. Wenn kein Papier mehr eingespannt ist, erfolgt eine akustische Warnung, und der Druck wird gestoppt.

Pica
Im Gegensatz zu »Elite« bezeichnet »Pica« eine Zeichendichte von 10 cpi.

Spooler
Ein Drucker-Spooler speichert Daten, die vom Rechner kommen, aber vom Drucker noch nicht verarbeitet werden können. Dadurch ist der Computer für andere Aufgaben frei und muß nicht auf den langsamen Drucker warten.

Unidirektional
Im »unidirektionalen« Modus wird im Gegensatz zum bidirektionalen nur in eine Richtung gedruckt, nämlich von links nach rechts.

Übersicht der IBM-Proprinter-Steuerbefehle

Befehl	Funktion
chr$(7)	Druckersignal
chr$(8)	Druckkopfrückschritt
chr$(9)	nächster Horizontaltabulator
chr$(10)	Zeilenvorschub
chr$(11)	nächster Vertikaltabulator
chr$(12)	Blattvorschub
chr$(13)	Druckkopf in Ausgangsposition
chr$(14)	Breitdruck für eine Zeile ein
chr$(15)	Schmaldruck ein
chr$(17)	Online-Status
chr$(18)	Schmaldruck aus oder...
chr$(18)	... Pica Ausdruck ein
chr$(20)	Breitdruck für eine Zeile aus
chr$(24)	letzte Zeile löschen
chr$(27)	Escape-Befehl
chr$(127)	letztes Zeichen löschen (Backspace)
<Esc>"-"0	Unterstreichen aus
<Esc>"-"1	Unterstreichen ein
<Esc>"0"	Zeilenabstand 1/8 Zoll
<Esc>"1"	Zeilenabstand 7/72 Zoll
<Esc>"2"	Zeilenabstand von <Esc>"A" ausführen
<Esc>"3"n	Zeilenabstand n/216 Zoll
<Esc>"4"	Blattanfang auf aktuelle Druckkopfposition
<Esc>"5"0	Wagenrücklauf ohne automatischen Zeilenvorschub
<Esc>"5"1	Wagenrücklauf mit automatischem Zeilenvorschub
<Esc>"6"	IBM Zeichensatz 2 ein
<Esc>"7"	IBM Zeichensatz 1 ein
<Esc>":"	Elite-Ausdruck ein
<Esc>"A"n	Zeilenabstand n/72 Zoll
<Esc>"B"n1 n2...0	vertikale Tabulatorpositionen setzen
<Esc>"B"0	vertikale Tabulatorpositionen löschen
<Esc>"C"n	Seitenlänge in n Zeilen
<Esc>"C"0n	Seitenlänge in n Zoll
<Esc>"D"n1 n2...0	horizontale Tabulatorpositionen setzen
<Esc>"D"0	horizontale Tabulatorpositionen löschen
<Esc>"E"	Fettdruck ein
<Esc>"F"	Fettdruck aus
<Esc>"G"	Schattendruck ein
<Esc>"H"	Schattendruck aus
<Esc>"J"n	einmaliger Zeilenvorschub mit n/216 Zoll
<Esc>"K"n1 n2 m1 m2...	Grafik in Normaldichte
<Esc>"L"n1 n2 m1 m2...	Grafik in Doppeldichte
<Esc>"N"n	unteren Rand setzen
<Esc>"O"	unteren Rand löschen
<Esc>"R"	horizontale und vertikale Tabulatoren auf Standardwert (gewöhnlich acht) setzen
<Esc>"S"0	Superscript ein
<Esc>"S"1	Subscript ein
<Esc>"T"	Super- oder Subscript aus
<Esc>"W"0	Breitdruck aus
<Esc>"W"1	Breitdruck ein
<Esc>"Y"n1 n2 m1 m2...	Grafik in Doppeldichte und doppelter Geschwindigkeit
<Esc>"Z"n1 n2 m1 m2...	Grafik in Vierfachdichte
<Esc>"\"n1 n2	Ausdruck aller verfügbaren Zeichen
<Esc>"^"n	Ausdruck eines Zeichens aus allen verfügbaren Zeichensätzen
<Esc>"_"0	Überstreichen aus
<Esc>"_"1	Überstreichen ein

Die IBM-Proprinter-Steuerbefehle

Weil die meisten Drucker über die IBM-Proprinter-Emulation verfügen, sind in der Tabelle alle Steuerbefehle abgedruckt, die der IBM-Proprinter versteht. Allerdings können einige Drucker nicht alle Steuerzeichen verarbeiten, so daß Sie in diesem Fall in Ihrem Druckerhandbuch nachschauen müssen. (wn)

BATH BEACH
SOFTWARE-SERVICE

PD/USA – VORSPRUNG MACHT HAPPY

PUBLIC DOMAIN 130

Aktuelles PD-Paket:
Insgesamt 130 Programme für **DM 129,00**

Translator Übersetzt Englisch-Programm-Dokumentation in Deutsch! **Lightning Press** DTP-Desktop-Publishing. Fantastic **As Easy As** Sehr gutes Tabellenkalkulationsprogramm. Bestseller **Prophet Lotto** Statistikprogramm für Lotterieziehungen **3D Chess** Dreidimensionales Schachprogramm **PC Write** Textverarbeitung mit vielen Möglichkeiten, sehr gut **Free File** Datenverwaltung mit vielen Möglichkeiten, sehr gut **Composer** Musik komponieren, Noten schreiben und ausdrucken, sehr gut **PC Akt** Erzeugt digitalisierte Aktbilder auf Ihrem PC **Dancad 3D** Dreidimensionales CAD-Zeichenprogramm mit Animation **PC KWIK** Beschleunigt Ihr Diskettenlaufwerk **Mahjongg** Chinesisches Spiel, sehr gut, mit guter Grafik **Megagopoly** Wie erreiche ich eine Million? Spielen Sie es! **Nuke New York** Zeigt einen nuklearen Angriff und die Auswirkung auf New York **Designer** Zeichnungsprogramm, erstellt bewegliche Bilder **Black Jack** Black Jack wie im Casino **Horoskop** Erstellt Ihr persönliches Horoskop **Kniffel** Kartenspiel **Mastermind** Populäres Strategiespiel **Star Traders** Weltraum-Händlerspiel. Aktien kaufen im Weltraum! **Etikett** Etiketten in verschiedenen Formaten erstellen und drucken **Flugsimulator** Flugsimulator **PC Calc** Populäre Tabellenkalkulation **Copy All** Nun können Sie die meisten Programme kopieren **PC Window** PC-Fenstertechniken **Kalender** Erstellt einen Kalender für jedes Jahr mit Bildschirm/Druck **Printum** Schaltet Drucker in Schriftarten um **Bastools** Basic-Tools **Your Menu** Erstellt Ihr eigenes Menü zum Ausführen **Form Master** Umfangreicher Formulargenerator **DOS Help** Alle DOS-Befehle werden gezeigt und erklärt **Stock Charting** Aktienverwaltung, Kauf-/Verkaufssignalgrafik **Biorythm** Biorythmen **Finance** Finanzen und Berechnungen **Taskplan** Zeitplanungsprogramm **Cover** Diskettenhüllen mit Inhaltsverzeichnis drucken **VCR Base** Videokassetten verwalten und Aufkleber drucken **Address** Adressen verwalten und Aufkleber erstellen **Bookinventory** Inventurprogramm für bis zu 2000 Bücher **PC Lock** Schützt Ihre Festplatte mit Password **Best Utilities** Eine Menge Hilfsprogramme für jeden etwas **Cataloging** Diskettenkatalog erstellen **FB Copy** Kopierprogramm, sehr schnell und komfortabel **Virus Check** Computerviren-Untersuchungsprogramm **Backgammon** Backgammonspiel **Space Invaders** Bekanntes Arcadespiel, sehr schön **PC Foto** Fotoverwaltungsprogramm und Diskaufkleber drucken **Packdisk** Macht mehr Platz auf Ihrer Festplatte **Compress** Datenkompakter, spart Platz auf der Diskette **Chasm** Super Assembler-Programmiersprache **plus weitere 90 Public Domain- und Shareware-Programme.**

Hilfs-, Anwendungs- und Spielprogramme (.EXE, .COM und .BAS-Start-Dateien) für IBM PC/XT/AT mit EGA-, CGA- und Hercules-Karte. Lieferung nur auf 5,25" Disketten. Schnellversand am Tag des Bestelleingangs. **Info-Katalog** mit über 1.500 Disketten-Beschreibungen und zusätzlichen Angeboten liegt jeder Bestellung bei.

Bath Beach Software-Service
Speckstraße 23, D-6749 Schweighofen, Telefon (06342) 7932

Bestell-Coupon

Ja, bitte senden Sie mir umgehend Ihr aktuelles PD-Paket „Public Domain 130" zum Discount-Preis von DM 129,00. Den Gesamtbetrag (Endpreis) bezahle ich wie angekreuzt. (Ausland nur gegen Vorauskasse!)

❑ Scheck/bar + DM 6,00 Versandkosten (Endpreis: **DM 135,00**)
❑ Per Nachnahme + DM 10,00 Versandkosten (Endpreis: **DM 139,00**)

Name:
Straße:
PLZ/Ort:
Telefon:
Datum/Unterschrift Geburtsdatum
(Bei Minderjährigen der Erziehungsberechtigte)

DOS Extra 10'90

Datenkompression – So klein mit Hut

Wenn lange Dateien kürzer werden, ohne daß Informationen verlorengehen, dann handelt es sich um Datenkompression. »Packen«, »Quetschen« oder gar »Zermalmen« lauten die Übersetzungen der englischen Ausdrücke für die verschiedenen Kompressionsmethoden. Im folgenden lesen Sie, was dahintersteckt.

Es ist natürlich nicht immer sinnvoll, Dateien zu komprimieren. Zum einen leidet der schnelle Zugriff darunter, wenn Dateien komprimiert sind, zum anderen ist der PC mitunter eine ganze Weile damit beschäftigt, die Dateien zu stauchen. Die Dateiverkleinerung ist jedoch sinnvoll im Zusammenhang mit Datenübertragungen, die Kosten verursachen, oder bei Datenarchiven, die keinen unmittelbaren Zugriff auf das Material verlangen, dafür aber platzsparend sein sollen.

Angenommen, Sie haben eine Datei von beispielsweise 500 KByte Länge. Diese Datei möchten Sie einem Freund zukommen lassen. Da Sie mit 360-KByte-Disketten arbeiten, wäre es Ihnen lieb, wenn die Datei auf weniger als 360 KByte schrumpfen würde. Oder Sie wollen Ihrem Kollegen die Datei per DFÜ (Modem, Akustikkoppler) schicken – je kürzer die Datei, desto geringer die Telefonkosten. Dafür darf Ihr Prozessor einige Zeit damit beschäftigt sein, dieses Datenpaket zu schnüren.

Ein weiterer Vorteil dieser »Archivprogramme«, wie sie auch genannt werden, ist, daß sie auch mehrere Dateien in eine solche Archivdatei hineinpacken. Damit sind ganze Anwendungen mit Hilfs- und Konfigurationsdateien unmißverständlich zu bündeln und zu verkleinern (Bild 1).

Es dürfte klar geworden sein, daß es Unsinn ist, alle Dateien Ihrer Festplatte nun etwa stundenlang zu komprimieren, um sie bei Bedarf wieder aufwendig zu dekomprimieren. Zum einen empfiehlt sich das Verfahren nur bei etwas größeren Dateien und zum anderen auch bei wenig gebrauchten Dateien, die Sie nicht auslagern wollen.

Bemerkenswert ist, daß die meisten Programme, die sich auf das Verkleinern von Originaldateien spezialisiert haben, in der Public-Domain- oder Shareware-Szene gewachsen sind. Bekannte ältere Namen sind etwa »pkpak« von Pkware, »arc« von System Enhancement Associates oder das Paar »arca« und »arce« von Wayne Chin und Vernon Buerg. Zu den neueren und leistungsfähigeren Programmen gehören »pkzip« von Pkware, »zoo« von Rahul Dhesi und »lharc« von Haruyasu Yoshizaki. Dazu gesellen sich ferner komfortable Benutzeroberflächen wie »arcmaster«, »idcshell« oder »shez«, die unter anderem sogar Einblick in diese archivierten Dateien gewähren.

Wie gehen Sie bei der Kompression vor? Sie wählen zum Beispiel eines der erwähnten Kompressionsprogramme und komprimieren die Datei, wobei Sie eine Vielzahl von Optionen einsetzen können. Neuere Programme beherrschen sogar mehrere dieser Kompressionsverfahren (Bild 2). Wenn Sie Glück haben, schrumpft die Datei dabei ganz beträchtlich (Bild 3). Haben Sie allerdings Pech, dann bringt die ganze P(l)ackerei kaum zehn Prozent Stauchung. Womit die Funktionsweise der Kompressionsprogramme in den Mittelpunkt des Interesses rückt.

Einfach, aber manchmal wirkungsvoll: die Codierung der Wiederholungsanzahl

Die einfachste Methode der Datenkompression ist das sogenannte »Packing« oder »Run-Length-Encoding« (Codierung der Wiederholungsanzahl). Wenn es sich bei Ihrer Datei beispielsweise um formatierten Text handelt, dann können am Anfang oder am Ende jeder Zeile unter Umständen eine Menge Leerzeichen stehen. Diese vielen gleichen Zeichen lassen sich kürzer darstellen. Stehen in der Originaldatei etwa 20 Leerzeichen, also 20 Byte, hintereinander, dann werden daraus in der Kompression drei Byte: Das erste Byte gibt an, um welches Zeichen es sich handelt. Das zweite Byte ist ein Flag-Byte. Es signalisiert, daß hier komprimiert wurde, und das dritte Byte sagt aus, wie oft das Zeichen sich wiederholt.

Es leuchtet ein, daß diese Methode nur dann etwas bringt, wenn mehr als drei gleiche Zeichen aufeinanderfolgen. Das kommt in Textdateien kaum vor, selbst die Leerzeichen werden von den Textprogrammen normalerweise nicht wiederholt, sondern in Form von Tabulatoren und Zeichen für Zeilenenden gespeichert. Packing wird also bei Textdateien kaum erfolgreich sein.

Bei binären Dateien, also Grafiken oder Zeichensätzen, sieht das schon ganz anders aus. Das Schöne beim Komprimieren einer Binärdatei ist, daß Sie nicht festzuhalten brauchen, welches Zeichen gerade komprimiert

C:\TEM

Name		Größe	Datum	Zeit
..		↑UP←DIR→	18.04.89	16:31
arca	com	4893	18.09.86	3:26
arce	com	7136	16.09.87	3:12
arc	exe	32512	16.12.88	14:59
lharc	exe	31256	31.05.89	11:30
narc	exe	28215	16.12.88	15:00
pak	exe	68176	4.02.89	23:10
pkpak	exe	21720	16.12.88	15:00
pkunpak	exe	15112	16.12.88	15:00
pkunzip	exe	21440	21.07.89	1:01
pkzip	exe	31342	21.07.89	1:01
shez	exe	40533	23.02.89	13:33
zoo	exe	40032	16.12.88	15:00
xompr	lzh	233771	21.07.89	1:01
lharc	man	40918	31.05.89	11:30

382.477 Bytes in 13 ausgew. Dateien

C:\TEM>

Bild 1. Die mit »Lharc« komprimierte Datei verbraucht nur noch 61,1 Prozent des ursprünglichen Speicherplatzes – da lohnt es sich doch zu komprimieren

```
Syntax for PAK is:
PAK <command> [/opt] [/opt..] <archive name> [file] [file...]

Commands:
 A  Add files to archive              M  Move files to archive
 U  Update archive files              F  Update duplicate files
 E  Extract files from archive        X  Move files from archive
 D  Delete files                      L  List files
 V  List files                        P  Display files
 T  Test files                        C  Convert files
 R  Revise remarks                    H  Revise header remark

Options:
 C   use Crunching compression        S   use Squashing compression
 G   Garble with password             M   Move files
 R   with Remarks                     H   With archive header remark
 D   Duplicate files only             P   Pack archives
 OM  Sort by Name                     OT  Sort by time
 OS  Sort by Size                     OE  Sort by extension
 O-  no sort                          OA  Write over Always
 OP  Prompt Write over                OO  Write over Older
 ON  Never Write over                 L   Use last date in archive
 T   set Temporary path
```

Bild 2. Moderne Kompressionsprogramme verstehen viele Optionen und beherrschen mehrere Arten des »Packens«, zum Beispiel Squashing oder Crunching

```
C:\TEM>pkzip -a komp

PKZIP (tm)   FAST!   Create/Update Utility   Version 1.01   07-21-89
Copyright 1989 PKWARE Inc.   All Rights Reserved.   PKZIP/h for help

Creating ZIP: KOMP.ZIP
  Adding: LHARC.EXE      imploding (37%), done.
  Adding: KOMPR.LZH      storing   ( 0%), done.
  Adding: ARC.EXE        imploding (47%), done.
  Adding: ARCA.COM       imploding (27%), done.
  Adding: ARCE.COM       imploding (32%), done.
  Adding: LHARC.MAM      imploding (67%), done.
  Adding: MARC.EXE       imploding (31%), done.
  Adding: PAK.EXE        imploding (47%), done.
  Adding: PKPAK.EXE      imploding (26%), done.
  Adding: PKUNPAK.EXE    imploding (22%), done.
  Adding: PKUNZIP.EXE    imploding (24%), done.
  Adding: PKZIP.EXE      imploding (27%), done.
  Adding: SHEZ.EXE       imploding (41%), done.
  Adding: ZOO.EXE        imploding (38%), done.
  Adding: BILD8.EXE      imploding (59%), done.
  Adding: KOMP.ZOO       storing   ( 0%), done.

C:\TEM>
```

Bild 3. »Pkzip« drückt Ihre Dateien schon ganz ordentlich zusammen – nur die schon komprimierten Dateien (mit der Endung »lzh« und »zoo«) kann es nicht weiter verkleinern

wird. Es kommen ja nur zwei in Frage, die sich irgendwie immer abwechseln.
So hat also auch diese mehr oder weniger triviale Methode mitunter ihre Berechtigung. Die gebräuchlichen Kompressionsprogramme versuchen auch immer als erstes, durch Packing etwas zu erreichen. Erst danach gehen sie zu den komplizierteren Methoden über.

Platzsparen mit Codieren in variabler Länge – das Pferd vom Schwanz her aufzäumen

Eine Datei zu kürzen, die aus vielen verschiedenen Zeichen besteht, scheint auf den ersten Blick unmöglich. Schließlich können die Bits nicht dichter zusammengeschoben oder übereinandergestapelt werden. Und doch gibt es einen Weg. Er nennt sich »Squeezing« (Quetschen) oder auch »Variable-Length-Encoding« (Codieren in variabler Länge).
Normalerweise wird jedes Zeichen durch ein Byte repräsentiert. Das ergibt zum Beispiel bei dem Wort »Schubidubidu« zwölf Buchstaben mal 8 Bit, also 96 Bit. Die Buchstaben, die nur einmal vorkommen, verbrauchen genausoviel Platz wie die Buchstaben, die häufiger vorkommen. Wenn man hier ansetzt, kann eine Menge Platz gespart werden. Die beiden am häufigsten vorkommenden Zeichen werden jeweils mit nur einem einzigen Bit codiert, nämlich mit 0 und 1. Die nächsthäufigen Zeichen codieren Sie mit 2 Bit, also 00, 01, 10 und 11. Im Beispiel »Schubidubidu« kommt das »u« am häufigsten vor, dann folgen »b«, »d« und »i«. So wird also »u« als 0 codiert.
Die folgenden drei Buchstaben treten alle mit gleicher Häufigkeit auf, nämlich zweimal: »b« wird 1, »d« wird 00 und »i« wird 10. Es bleiben noch die Buchstaben, die nur einmal vorkommen: »c« wird 11, »h« wird 000, »S« wird 001. Das Ergebnis sieht dann folgendermaßen aus:

001 11 000 0 1 10 00 0 1 10 00 0

Aus 96 Bits sind 21 geworden – beeindruckend fürwahr! Werden jedoch die Leerzeichen weggelassen, wissen weder Mensch noch Computer mehr, wo der Code für ein Zeichen aufhört und der Code für das nächste beginnt. Das bedeutet, daß entweder Trennzeichen gebraucht werden oder aber die Zeichen so codiert werden müssen, daß keiner der kurzen Codes dem Anfang eines längeren Codes entspricht. Trennzeichen sind jedoch keine gute Methode, denn dann wäre die komprimierte Datei am Ende länger als die Originaldatei.

Mit der Huffman-Codierung lassen sich auch höchst uneinheitliche Dateien stark verkleinern

Das Verfahren des Huffman-Coding (nach dem Erfinder des Algorithmus benannt) beschreibt denn auch eine eindeutige Codierung der Zeichen in variabler Länge und zwar ausgehend von der Häufigkeit.
Der erste Schritt besteht demnach wieder darin, zu zählen, wie oft jedes Zeichen vorkommt. Dazu erhält jedes Zeichen zunächst einen Index, mit dem es sich leichter als mit den originalen Zeichen arbeiten läßt. Dann wird die Datei nach jedem Zeichen durchsucht und der jeweilige Zähler erhöht, sooft das entsprechende Zeichen gefunden wird (Bild 4). Danach ist bekannt, wie oft jedes Zeichen vorkommt. Bild 5 zeigt das Ergebnis bei dem Satz »Datenkompression ist keine Hexerei«.

```
for i:=0 to 256 do zaehler[i]:=0;
for i:=1 to max do
  zaehler[Index(b[i])]:=succ(zaehler[Index(b[i])]);
```

Bild 4. Diese Routine stellt fest, wie oft jedes Zeichen vorkommt

Nun geht es ans Codieren. Das Prinzip der Huffman-Codierung ist grafisch sehr gut zu demonstrieren: Stellen Sie sich für jedes vorkommende Zeichen ein Kästchen vor, in dem vermerkt ist, wie häufig dieses Zeichen auftritt. Dies sehen Sie in der ersten Reihe von Bild 6.
Jetzt werden die Zeichen gesucht, die am seltensten vorkommen. Im vorliegenden Beispiel kommen sieben Zeichen nur einmal vor. Sechs dieser Zeichen werden zu Zweiergruppen gebündelt, und das siebte wird mit einem zweimal vorkommenden Zeichen zusammengefaßt. Diese vier Zweiergruppen erhalten jetzt jeweils einen Knotenpunkt, der den Wert erhält, der der Summe seiner beiden Zweige entspricht (Bild 6, zweite Reihe).
Nun werden erneut die Elemente mit den geringsten Werten zusammengefaßt. Das können einzelne Kästchen oder auch Knotenpunkte mitsamt ihren Verzweigungen sein. (Bild 6, dritte Reihe). Dieses Verfahren wird so lange fortgesetzt, bis alle Zeichen an einem Baum »hängen«. In der letzten Reihe von Bild 6 sind die Zeichen noch einmal dargestellt.
Dieser Baum wird jetzt als Codierungsanweisung betrachtet. Jede Verzweigung nach links entspricht einer 0 und jede Verzweigung nach rechts einer 1. Sind nun alle Forderungen, die an die Codierung gestellt werden, erfüllt? Werden die am häufigsten vorkommenden Zeichen am kürzesten codiert?
Die Antwort lautet »Ja«. Sie sehen, daß die am häufigsten auftretenden Zeichen im Baum am höchsten stehen. Man benötigt weniger Verzweigungen, um zu ihnen zu kommen. Und da jede Verzweigung ein Bit darstellt, besitzen also die häufigsten Zeichen die kürzesten Codes. Das »e« ist zum Beispiel mit zwei Bit – nämlich 01 – codiert.
Ist kein kürzerer Code der Anfang eines längeren Codes, so daß jeder Code eindeutig definiert ist? Dies trifft ebenfalls zu. Man kann es am Baum einfach ablesen, denn kein Zeichen steht in direkter Linie über ei-

Zeichenhäufigkeitstabelle für Huffman-Codierung

Zeichen:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	.
Häufigkeit:	1	0	0	1	7	0	0	1	4	0	2	0	1	3	2	1	0	2	3	2	0	0	0	1	0	0	3

Bild 5. Der Einfachheit halber wurden in dieser Darstellung nur die Buchstaben des Alphabets und das Leerzeichen berücksichtigt

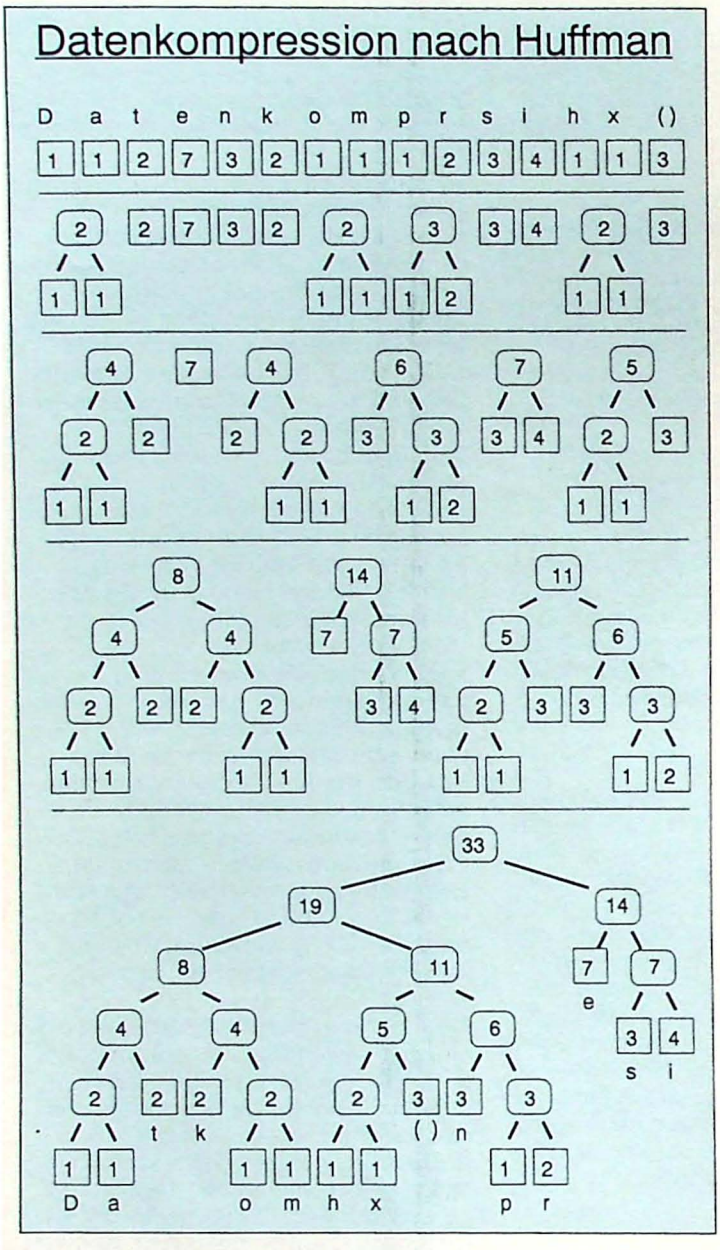


Bild 6. Hier sehen Sie Schritt für Schritt, wie der Huffman-Baum konstruiert wird

```
Codierungstabelle bei Huffman:

E:  01
S:  001   I:  000
N:  1001  S:  1010  K:  1101  T:  1110
R:  10000 P:  10001 X:  10110 H:  10111
M:  11000 O:  11001 A:  11110 D:  11111
```

Bild 7. So entsteht das Initial-Array für die Konstruktion des Huffmann-Baums

nem anderen Zeichen, mit dem es verwechselt werden könnte. In Bild 7 ist die Codierung der Buchstaben des Satzes »Datenkompression ist keine Hexerei« abgedruckt.

Ihnen ist sicher aufgefallen, daß es jetzt kein Zeichen mehr gibt, das mit nur einem Bit codiert ist. Das ist der Preis, der für die Eindeutigkeit der Codierung bezahlt werden muß. Ganz so »optimistisch« kurz wie im vorherigen »Schubidubidu-Beispiel« geht es also doch nicht.

Nachdem nun alles seine Richtigkeit hat, können Sie sich an die Programmimplementierung machen. Das Array mit den Zeichen und ihrer Häufigkeit ist bereits angelegt (Bild 5). Aus diesem Array werden jetzt die Zeichen entfernt, die in der Datei nicht vorkommen. Dabei wird neu durchnumeriert (Bild 8).

Nun nimmt das Programm die beiden kleinsten Werte, also die Zeichen, die am seltensten vorkommen, aus dem Array heraus. Sie werden addiert und danach als Summe – versehen mit einem Index – in das Array zurückgepackt. Bei jedem Durchgang wird das Array also um ein Element kleiner.

Die Summe der beiden Werte ist natürlich der Wert ihres gemeinsamen Knotenpunkts im Baum. Die Position dieser Zeichen im Baum – und damit ihr Code – wird dadurch definiert, daß das Programm in einem weiteren Array festhält, welches der gemeinsame Knoten der beiden Zeichen ist. Zur Unterscheidung zwischen der rechten und der linken Verzweigung wird der Knoten bei der linken Verzweigung mit negativem Vorzeichen angegeben. Jetzt werden die zwei nächstniedrigen Werte gesucht und das gleiche Verfahren wird mit ihnen wiederholt. Bild 9 zeigt, wie dies gemacht wird. Jetzt kann endlich jedem Zeichen sein Code zugeordnet werden. Der Code wird durch zwei Arrays dargestellt.

Das erste, »code[k]«, enthält die binäre Repräsentation des k-ten Buchstabens, das zweite Array »laenge[k]« gibt an, wie viele Bits der Code lang ist (Bild 10). Zu guter Letzt nimmt das Programm diesen Code und verschlüsselt damit die Datei (Bild 11).

Bild 8. Das sind die Huffman-Codes der einzelnen Zeichen des Beispielsatzes

```
procedure EntfElement(k: integer);
 label Abbruch;
 var j,v: integer;
begin
  v:=p[k];
  while (k<=n shr 1) do
  begin
    j:=k+k;
    if (j<n) then
      if (b[p[j]]<b[p[succ(j)]]) then j:=succ(j);
    if (b[v]>=b[p[j]]) then goto Abbruch;
    p[k]:=p[j];
    q[p[j]]:=k;
    k:=j
  end;
  Abbruch:
  p[k]:=v;
  q[v]::=k
end; {EntfElement}

n:=0;
for i:=0 to 256 do
  if (zaehler[i]<>0) then
  begin
    n:=succ(n);
    heap[n]:=i
  end;
for k:=n downto 1 do EntfElement(k);
```

```
repeat
  t:=heap[1];
  heap[1]:=heap[n];
  n:=pred(n);
  nh:=256+n;
  EntfElement(1);
  zaehler[nh]:=zaehler[heap[1]]+zaehler[t];
  vorg[t]:=nh;
  vorg[heap[1]]:=-nh;
  heap[1]:=nh;
  EntfElement(1)
until (n=1);
vorg[nh]:=Ø;
```

Bild 9. Hier entsteht die Knotenrepräsentation des Huffman-Baums

Wer Lust hat, kann jetzt ausrechnen, wie lang der Satz »Datenkompression ist keine Hexerei« nach der Kompression ist. Dazu brauchen Sie nur für jedes Zeichen die Anzahl der Verzweigungen aus Bild 6 zu zählen und die Zahl mit der Häufigkeit des Vorkommens zu multiplizieren. Danach werden die Summen addiert. Dabei kommt dann heraus, daß der Satz statt der vorherigen 272 Bit (34 Zeichen x 8 Bit) jetzt nur noch 105 Bit lang ist.

Doch eines sollte nicht übersehen werden: Das ganze Codieren nützt ziemlich wenig, wenn am Ende die Häufigkeitstabelle nicht mehr vorliegt. Das heißt, die komprimierte Datei kann nicht nur aus den kürzer codierten Zeichen bestehen, sondern muß auch diese Chiffriertabelle enthalten. Wieviel Platz das nun wieder kostet, hängt hauptsächlich davon ab, wie viele verschiedene Zeichen vorkommen. Wenn volle 256 Zeichen codiert werden müssen, nimmt die Chiffriertabelle rund 1 KByte ein. Das bedeutet, daß die Huffman-Codierung nur bei Dateien von mehr als rund 5 KByte Größe Sinn macht. Aber wer will schon kurze Dateien kürzen?

Der Huffman-Code macht sich auch im Telefaxdienst nützlich. Faxgeräte enthalten eine Abwandlung dieses Codes, genannt MH (Modified Huffman), der die Datenübertragung beschleunigt.

Den Herren Lempel, Ziv und Welch waren ihre auf diese Art komprimierten Dateien immer noch zu umfangreich. Ihr LZW-Algorithmus, auch als »Crunching« (Zermalmen) bekannt, besitzt zwei entscheidende Vorteile gegenüber dem Huffman-Code: Erstens ist es nicht mehr notwendig, die komprimierte Datei durch eine Codierungstabelle zu verlängern. Zweitens werden nicht nur einzelne Buchstaben, sondern gleich ganze Zeichenketten mit einem kurzen Code versehen. Auch der Kompressionsvorgang wird beschleunigt. Während bei Huffman jede Datei zweimal durchgearbeitet werden muß – einmal, um die Häufigkeit der Zeichen festzustellen, das zweite Mal zum Codieren – geschieht hier alles auf einmal.

Beim Crunching wird im Speicher eine Tabelle angelegt, die zunächst aus 256 Einträgen für die möglichen Zeichen besteht. Für jedes Zeichen

```
for k:=Ø to 256 do
  if (zaehler[k]=Ø) then
  begin
    code[k]:=Ø;
    laenge[k]:=Ø
  end;
  else begin
    i:=Ø;
    j:=1;
    t:=vorg[k];
    x:=Ø;
    repeat
      if (t<Ø) then
      begin
        x:=x+j;
        t:=-t
      end;
      t:=vorg[t];
      j:=j+j;
      i:=succ(i)
    until (t=Ø);
    code[k]:=x;
    laenge[k]:=i
  end;
```

Bild 10. Der Code jedes Zeichens wird durch zwei Arrays bezeichnet

sind drei Byte vorgesehen: zwei Präfixbytes und ein Suffixbyte, das den ASCII-Wert des jeweiligen Zeichens enthält.

Jetzt geht das Programm die Datei durch, die komprimiert werden soll. Bei dem Satz »Datenkompression ist keine Hexerei« stößt es zuerst auf ein »D«. Dieses »D« wird in der Tabelle gesucht. Die Adresse des »D« aus der Tabelle wird in die Ausgabedatei übernommen. Der zweite Buchstabe ist ein »a«. Jetzt sieht das Programm nach, ob die Zeichenkette »Da« schon in der Tabelle eingetragen ist. Weil das nicht der Fall ist, wird ein neuer Eintrag vorgenommen. Das Präfix dieses Eintrags ist ein Zeiger, der auf den Eintrag des »D« zeigt. Das Suffix ist der ASCII-Wert des »a«. Damit ist also »Da« als Zeichenkette in der Tabelle festgehalten und kann, im Wiederholungsfall, in der Ausgabedatei mit einem Code dargestellt werden. Die Zeichenkettentabelle verändert sich also während der Komprimierung.

Noch hat das Crunchen nichts gebracht, denn die Codierung für die Buchstaben »Da« ist länger, als dies der Fall wäre, wenn gar nichts getan worden wäre. Aber bei großen Dateien, wo sich längere Zeichenketten zwangsläufig wiederholen, ist dieses Verfahren sehr effektiv.

Für die Methode von Lempel, Ziv und Welch interessieren sich übrigens nicht nur speicherplatzgeplagte Computerbenutzer, sondern auch die CCITT (Comité Consultatif International Télégraphique et Téléphonique), das beratende Organ der Internationalen Fernmeldeunion. Wahrscheinlich wird dieser Algorithmus als Standard für die Datenkompression in Modems zugelassen. Das heißt, daß Modems nach dieser Methode selbsttätig Dateien komprimieren werden, bevor sie diese übertragen. Aber bis dahin müssen Sie noch selbst komprimieren – sei es um Telefonkosten zu senken, Disketten zu sparen oder Platz auf der Festplatte zu schaffen.

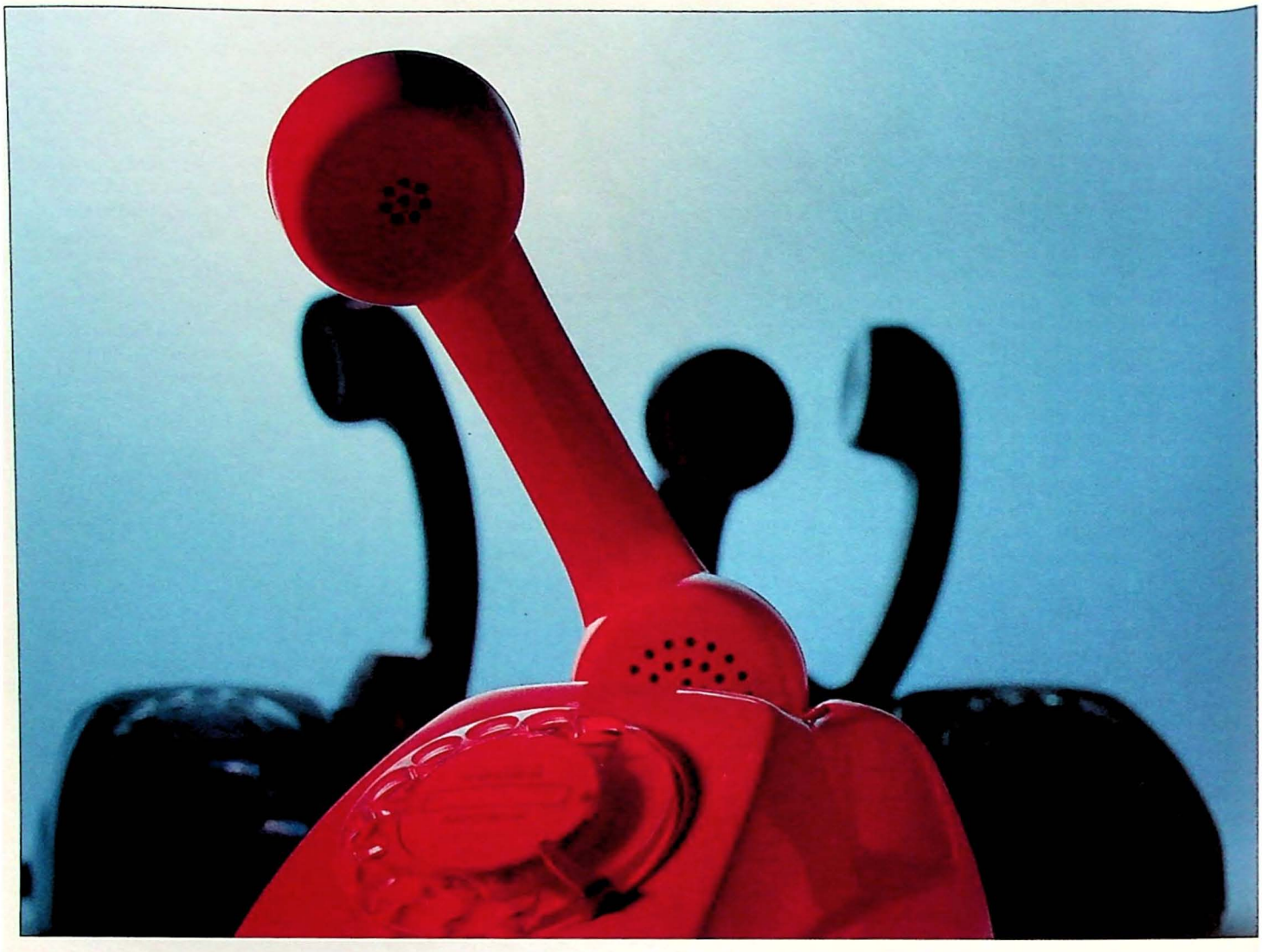
(Fredrika Gers/Stefan Willmeroth/al)

Info: PD-Service-Lage, 4937 Lage/Lippe

Komprimierprogramm	Bestellnummer
Lharc	A379
Pak	A321
Pkzip	A322
Arc	A319
Oberfläche	**Bestellnummer**
Arcmaster	A155
Idcshell	A121
Shez	A605

```
function Bits(x,k,j: integer): integer;
 var erg: integer;
begin
  erg:=(x shr k) and pred(1 shl j);
  Bits:=erg
end;

for j:=1 to max do
  for i:=laenge[Index(b[j])] downto 1 do
    write(Bits(code[Index(b[j])],pred(i),1):1);
```

Bild 11. Jedem Zeichen wird ein Code zugeordnet



Tricks aus der Software-Hotline

Probleme ergeben sich am Rechner gar viele, sei es beim Programmieren oder sei es bei der täglichen Arbeit mit der Anwender-Software. Deshalb haben wir für Sie jede Menge Seiten mit Tricks und Kniffen, beispielsweise zu den Bereichen Programmiersprachen, Betriebssystem und Software, gefüllt.

Anwender-Software

Works

☐ *Kann ich Texte, die ich mit Works geschrieben habe, in das Format anderer Textverarbeitungsprogramme konvertieren?*

Ja. Es gibt ein Konvertierungsprogramm »wptowp.exe«, mit dem Sie Dateien in folgende Formate konvertieren können:
- Microsoft Interchange Rich Text Format (RTF)
- IBM Document Content Architecture Format (DCA)

Die Diskette kann bei Microsoft in der Support-Abteilung angefordert werden. Für die Korrektheit des Programms können wir keine Garantie übernehmen.

☐ *Gibt es einen Überschreibemodus in der Works-Textverarbeitung?*

Sobald Sie einen Text markieren, wird dieser mit den nächsten Tastenanschlägen überschrieben.

Word 5.0

☐ *Kann ich mit Word und einem Scanner Formulare ausfüllen?*

Zunächst müssen Sie das Formular mit einem Text-Scanner einlesen, anschließend muß der Text von einer Texterkennungs-Software gelesen werden. Auf dem Markt stehen hierfür verschiedene Programme zur Verfügung. Probleme bestehen zum Teil noch bei der Erkennung bestimmter Schriftarten. Danach kann der Text – wie jeder andere, direkt mit Word erzeugte – bearbeitet werden. Sie können an den entsprechenden Positionen im Formular Markierungspunkte setzen, die dann beim Ausfüllen angewählt werden. Die Eingabe können Sie mit Hilfe eines Makros automatisieren.

☐ *Kann ich die deutsche oder englische Word-Version auch zusammen mit Sonderzeichen anderer Sprachen oder sogar mit anderen Zeichensätzen wie Kyrillisch oder Hebräisch kombinieren?*

Zum normalen Lieferumfang von Word gehören verschiedene fremdsprachige und mathematische Sonderzeichen. Eine genaue Übersicht finden Sie im Anhang des Handbuchs »Word zum Nachschlagen«. Darüber hinaus bieten verschiedene Firmen Zusatzprogramme mit anderen Zei-

chensätzen. Die Palette wird ständig erweitert. Derzeit stehen folgende Zeichensätze zur Verfügung: Mathematik, Noten, Bulgarisch, Griechisch, Hebräisch, Katakana, Kyrillisch, Lettisch, Litauisch, Polnisch, Portugiesisch, Rumänisch, Russisch, Serbokroatisch, Slowakisch, Tschechisch, Türkisch, Ungarisch, Usbekisch sowie ein Maya-Kalender.

Infos: PC Software-Service, 8000 München 2
Lang Computer Consulting, 7146 Tamm
Simon Verlag, 5300 Bonn 1

☐ *Für welche Datenformate und Textverarbeitungsprogramme stehen Konvertierungsprogramme zur Verfügung?*

Die Konvertierungsprogramme für das DCA- und das RTF-Format können Sie kostenlos von der Microsoft-Support-Abteilung erhalten. Bitte wenden Sie sich hierzu an die Hotline. Mit dem Programm können Sie Texte aus anderen Textverarbeitungsprogrammen ins Word-Format konvertieren beziehungsweise Word-Texte in andere Formate übertragen. Sie können von der Firma Datacom in 8000 München 70 das Programm »Keyword« für die Textverarbeitungsprogramme DEC WPS Plus (DX), Display Write (TXT), Multimate, Samna, Volkswriter, Wang IWP, Wordperfect, Wordstar und Wordstar 2000 verwenden. Sie sollten sich also – falls Sie eine Konvertierung für ein nicht genanntes Programm suchen – auch an den Hersteller dieses Programms wenden. Bitte beachten Sie, daß alle Programme ohne Garantie auf Korrektheit ausgeliefert werden. In einigen Fällen können die Formatierungen nicht vollständig übernommen werden.

☐ *Kann ich die Bildschirmausgabe in Word so einstellen, daß ich mit schwarzer Schrift auf weißem Hintergrund arbeiten kann?*

Sie können nur im Textmodus – nicht im Grafikmodus – schwarze Schrift auf weißem Hintergrund ausgeben. Wählen Sie dazu »Zusätze, Bildschirm, Text« und »Ausschnitt, Optionen, Hintergrundfarbe, Weiß«.

☐ *Ich besitze den IBM-Monitor 8503. Wie kann ich mit Word in den sechzehn Graustufen arbeiten, die mir damit zur Verfügung stehen?*

Normalerweise arbeitet dieser Bildschirm im Videomodus 7 und stellt dabei wie ein Monochrom-Monitor nur zwei Farben dar. Mit Hilfe des Eintrags »mode co80« in der »autoexec.bat« können Sie auf den Videomodus 3 umschalten. Anschließend werden 16 Farben ausgegeben, wie bei einem Farbbildschirm.

☐ *Wie kann ich beim Arbeiten mit Druckformatvorlagen die Hilfefunktion mit einer Tastenkombination aufrufen?*

Während der Arbeit mit Druckformatvorlagen rufen Sie die Hilfefunktion und das Lernprogramm mit der Tastenkombination [Alt-X-?] auf.

☐ *Wie kann ich ein wahlweises Trennzeichen mit den Funktionen »Suchen« oder »Wechseln« bearbeiten?*

Wenn Sie mit den Funktionen »Suchen« oder »Wechseln« ein wahlweises Trennzeichen bearbeiten möchten, müssen Sie dem Trennzeichen noch ein Control-Zeichen (†) voranstellen. Ihre Eingabe lautet also [Alt-94], [Alt-45].

☐ *Wie viele Tabulatoren kann ich in einer Zeile setzen?*

Sie können maximal 19 Tabulatoren setzen.

☐ *Wie kann ich meinen Drucker selbst an Word anpassen? Es gibt keinen passenden Treiber dafür.*

Sie können von Microsoft kostenlos das Zusatzprogramm »makeprd.exe« erhalten, mit dessen Hilfe Sie eigene Druckertreiber schreiben können. Bitte beachten Sie, daß dieses Programm ohne Garantie auf Korrektheit und ohne Anspruch auf technische Unterstützung ausgeliefert wird. Die notwendigen Informationen finden Sie in Informationsdateien auf der Diskette.

☐ *Wo muß im Serienbrief die Kopfzeile stehen?*

Auch im Serienbrief steht die Kopfzeile am Anfang, also noch vor der Angabe der Steuerdatei.

☐ *Kann ich in Steuerdateien für Serienbriefe Zeilenschaltungen, Absatzmarken und Seitenumbrüche einfügen?*

Zeilenschaltungen ([Shift-Enter]) und Seitenumbrüche ([Ctrl-Shift-Enter]) können in Steuerdateien verwendet werden. Absatzmarken ([Enter]) kennzeichnen das Ende von Datensätzen und dürfen daher ansonsten nicht in Steuerdateien enthalten sein.

☐ *Kann ich in Word einen Eintrag in einem längeren Eingabefeld korrigieren, oder muß ich alles neu schreiben, wenn ich mich vertippt habe?*

Sie können sich mit den Funktionstasten [F9] und [F10] in den Eingabefeldern vorwärts und rückwärts bewegen. Wie üblich können Sie dann Zeichen einfügen oder löschen.

☐ *Ich habe einen Laserdrucker, in dem die Seitenbeschreibungssprache »Prescribe« integriert ist. Damit habe ich ein umfangreiches Formular entwickelt, das ich in Word einbinden möchte. Wie kann ich das erreichen?*

Hierfür gibt es zwei Alternativen: Zum einen können Sie die Formulardefinition als Textbaustein ablegen und diesen nach Bedarf einfügen. Zum anderen können Sie die Definitionsdatei auch vor dem Start auf den Drucker kopieren. Anschließend rufen Sie im Text nur ein entsprechendes Makro auf. In beiden Fällen gilt, daß der Absatz mit den Prescribe-Befehlen mit Zeilenabstand »0« formatiert wird. Außerdem müssen Sie den internen »Cursor« des Laserdruckers wieder an den Anfang der Seite stellen.

☐ *Nachdem ich Word mit dem Parameter »/h« gestartet habe, arbeitet es immer im 43-Zeilen-Modus, auch wenn ich das Programm ohne Parameter starte. Wie bekomme ich wieder 24 Zeilen?*

Sie müssen Word mit dem Parameter »/m« starten. Dadurch schaltet das Programm wieder in die normale Bildschirmdarstellung zurück.

☐ *Worin bestehen die Unterschiede zwischen »Darstellungsform, Druckbild« und »Darstellungsform, Normal« im Menü »Zusätze«?*

Auf dem Bildschirm können maximal 80 Zeichen pro Zeile ausgegeben werden. Dabei sind die Zeichen immer gleich groß, egal ob Sie den Text mit großem oder kleinem Schriftgrad formatiert haben. Falls Sie mit kleinen Schriftgraden arbeiten, können in einer gedruckten Zeile aber mehr als 80 Zeichen stehen. Wenn Sie »Darstellungsform, Normal« wählen, erfolgt der Zeilenumbruch entsprechend dem Bildschirm, das heißt, jede Zeile ist maximal 80 Zeichen lang. Auf dem Bildschirm sind immer vollständige Zeilen sichtbar. Der Text umfaßt aber auf dem Bildschirm mehr Zeilen als er auf dem Papier bräuchte.

Wählen Sie »Darstellungsform, Druckbild«, so entspricht der Zeilenumbruch auf dem Bildschirm dem auf Papier. Damit Sie den gesamten Text einer Zeile lesen können, müssen Sie nach links beziehungsweise rechts rollen. Beim Erfassen des Textes ist die Einstellung »Normal« günstiger, vor dem Druck sollten Sie aber zur Kontrolle auf »Druckbild« umschalten.

☐ *Kann ich Datensätze aus dBase in eine Steuerdatei für Serienbriefe übernehmen?*

Sie können problemlos aus dBase eine Steuerdatei für Serienbriefe herstellen. Schreiben Sie dazu in dBase in etwa folgende Routine:

```
set talk off
set safe off
use adressen && Name der Datenbank
go top
```

```
set alte to test.txt && Name der Steuerdatei in Word
??"name;vorname"
do while .not. eof()
   ?name,";",vorname
   skip
enddo
close alternate
```

Ihre dBase-Datenbank wird hiermit die Inhalte der beiden Felder »Name« und »Vorname« als Serienbriefdaten für Word übergeben. Sollten Sie andere Felder oder nur bestimmte Datensätze auswählen wollen, sind umfangreichere Programme nicht zu umgehen.

☐ *Wie kann ich verhindern, daß zwei Worte, die in der gleichen Zeile stehen müssen, durch einen Zeilenumbruch getrennt werden?*

Geben Sie zwischen den beiden Worten ein geschütztes Leerzeichen ein. Sie erhalten es durch die Tastenkombination [Ctrl-Leertaste].

☐ *Ich verwende die Rahmenfunktion und habe die Beschriftung innerhalb des Rahmens mit Pica 7 formatiert. Wieso sind die Linien auf dem Papier im Vergleich zu denen auf dem Monitor verschoben?*

Die Rahmenlinien werden immer im Schriftgrad 12 gedruckt. Wenn Sie für die Beschriftung einen anderen Schriftgrad wählen, kommt es zu Verschiebungen.

☐ *Wenn ich einen Absatz einrahme, geht der Rahmen über die gesamte Seitenbreite. Wie kann ich ihn verkleinern, um zum Beispiel ein zentriertes Wort einzurahmen?*

Die Breite des Rahmens richtet sich nach den linken und rechten Einzügen, die Sie unter »Format, Absatz« festlegen.

☐ *Wie kann ich in Word eine Grafik aus Excel einbinden?*

In Word können Grafiken nur als Druckdateien eingebunden werden, das heißt, die Datei muß sämtliche Informationen für die Druckaufbereitung enthalten. Excel ist eine Windows-Applikation. Wie Sie unter Windows eine Druckdatei erzeugen können, ist im Windows-Handbuch beschrieben.
Excel bedruckt generell eine ganze Seite. Daher müssen Sie alle Druckersteuersequenzen am Anfang und am Ende der Druckdatei löschen. Fügen Sie in Excel unter »Datei, Layout« am Anfang Ihrer Kopfzeilendefinition und am Ende Ihrer Fußzeilendefinition eine auffällige Zeichenkombination ein, und speichern Sie diese Datei als Druckdatei.
Laden Sie diese Druckdatei in Word, und löschen Sie alle Sonderzeichen vom Anfang der Datei bis zur Kopfzeile einschließlich Ihrer speziellen Merkzeichen. Ebenso löschen Sie die Zeichen am Ende Ihrer Datei und speichern anschließend die Datei unformatiert. Nun können Sie die Grafik aus Excel wie jede andere Druckdatei in Word einbinden.

☐ *Wie kann ich in Word einen Bereichswechsel suchen?*

Sie können einen Bereichswechsel mit der Funktion »Suchen« auffinden. Geben Sie dazu als Suchbegriff »^b« ein. Hier werden dann aber nicht nur Bereichswechsel, sondern auch Seitenwechsel gefunden. Um einen Bereich gezielt anzusprechen, geben Sie bitte folgendes Makro ein:

```
«Abfrage nr=?In welchen Bereich wollen Sie springen?»
                <umschalten f10><unt>fbp
<tab 3>b<return><unt>dun<return><unt>gb1b«nr»
<return>
```

Wichtig ist, daß die Seitennumerierung für jeden Bereich neu bei 1 beginnt. Durch die Eingabe des Befehls »Druck, Umbruch-Seite« wird der Text neu umgebrochen und in der Statuszeile die aktuelle Cursor-Position einschließlich der Bereichsnummer angezeigt.

Chart 3.0

☐ *Ich würde auch weiterhin gerne mit Chart die grafische Auswertung meiner Daten übernehmen. Allerdings bin ich von Multiplan auf Excel umgestiegen. Wie geht das?*

Starten Sie Chart, wählen Sie im Auflistungsbildschirm »XTERN/Multiplan« und gehen Sie von nun an genauso vor, wie Sie dies aus Multiplan gewohnt sind.

☐ *Ich besitze die Programme Windows, Pagemaker und Chart. Nun möchte ich eine Grafik, die ich in Chart hergestellt habe, über die Zwischenablage in Pagemaker einfügen. Wie gehe ich dabei vor?*

Starten Sie Chart wie gewohnt über eine Pif-Datei aus Windows. Lassen Sie in diesem Zusammenhang Chart nicht in einem Fenster ablaufen. Wenn Sie Ihre Grafik hergestellt haben, schalten Sie über das Menü »Zusätze« bitte auf »Negativdarstellung«. Jetzt drücken Sie die Tastenkombination [Alt-Druck]. Für einen Moment wird Ihr Bildschirm dunkel. Nun können Sie die Hardcopy des Bildschirms in Pagemaker einfügen. Dieser Vorgang hat den Vorteil, daß Sie eine bessere Auflösung der Grafik erzielen, als wenn Sie Chart im Fenster laufen lassen.

☐ *Wie muß ich Chart installieren, damit beim Start keine farbigen Säulen oder andere Darstellungen ausgegeben werden? Ich möchte die Grafiken nur in Schwarzweiß haben.*

Stellen Sie im Grafikbildschirm unter »Format, Vorgabe, Farb-, Tonwert« die Punkte »Fläche/Tonwert« auf »auto«, »Line/Tonwert« auf »auto« und »Datenreihe/-punkt« auf »Schwarz« ein. Danach stellen Sie unter »Format/Art« die Einstellung »Rubriken verschieden« auf »Ja«. Beenden Sie Chart. Bei einem erneuten Programmstart wird nun die Umsetzung der Daten in Schwarzweiß mit verschiedenen Mustern vorgenommen.

☐ *Ich habe eine Grafik-Show in Chart hergestellt. Nun kommt es vor, daß innerhalb eines Vortrags die Grafiken nicht nur in einer Reihenfolge, sondern manchmal auch vor und zurück geblättert werden sollen. Läßt sich dies realisieren?*

Nein. Sie können zwar mit »Ausliste« eine bestimmte Vorgehensliste anlegen, jedoch nicht die Reihenfolge individuell ändern.

☐ *Ich habe einen HP-Plotter 7475A. Nun habe ich im Chart-Handbuch gelesen, daß ich mit einer Stapeldatei ausdrucken könnte. Ich habe mich dabei genau an die Vorgaben gehalten, aber der Rechner fordert mich auf, bei »Druckerstift wechsel« mit »Ja« zu quittieren. Wie kann ich dies umgehen?*

Eigentlich fordert das Programm Sie nicht auf, die Plotter-Stifte zu wechseln, sondern ein neues Blatt Papier einzulegen. Daher können Sie mit Ihrem Plotter nicht mit einer Stapeldatei ausdrucken. Dies funktioniert nur mit Plottern, die über einen automatischen Papiereinzug verfügen.

☐ *Wenn ich Chart aus anderen Verzeichnissen heraus starte, werden immer die Initialisierungsdaten verwendet, die in der »mc.ini« im Programmverzeichnis von Chart stehen. Da ich aber mit unterschiedlichen Voreinstellungen arbeiten möchte, muß ich jedesmal die Einstellung ändern. Gibt es eine einfachere Lösung?*

Wenn Sie Chart mit dem Parameter »/n« starten – zum Beispiel C:

```
\chart\chart.com /n)
```

wird die »mc.ini« aus dem aktuellen Verzeichnis benutzt beziehungsweise für das aktuelle Verzeichnis erzeugt.

Excel

☐ *Ich habe eine Makro-Vorlage hergestellt und benutze darin die Funktion »Istzahl«. Diese Anweisung funktioniert jedoch nicht, wenn ich sie außerhalb eines Makros benutze. In einer Excel-Tabelle hingegen habe ich keine Schwierigkeiten. Was mache ich falsch?*

Makro-Vorlagen werden aus Geschwindigkeitsgründen im Gegensatz zu den Tabellen weder automatisch noch auf Befehl neu berechnet. Formeln und Funktionen werden erst dann neu berechnet, wenn sie Teil eines Befehls- oder Funktions-Makros sind und das jeweilige Makro gestartet wurde. Verwenden Sie in einer Makro-Vorlage deshalb Formeln und Funktionen nur innerhalb von Makros.

☐ *Wie kann ich in Excel zwei Zeitangaben (zum Beispiel 12:00 und 14:00) so summieren, daß dabei das Ergebnis 26:00 erzielt wird?*

Folgendes Beispiel verdeutlicht den Weg, um Zeiten wie gewünscht zu summieren:

```
A3:   12:30  Zeit in Std:Min
A4:    8:01  Zeit in Std:Min
A5:   12:35  Zeit in Std:Min
A6:   10:06  Zeit in Std:Min
A7:   43:12  Ergebnis der Addition
```

Feld A7 beinhaltet die Formel

```
=Wechseln(Fest(Ganzzahl(Summe(Stunde(A3:A6))+Summe(Minu-
 te(A3:A6))/60)+Rest(Summe(Minute(A3:A6))/60;1)*0,6;2)
 ;"";"";"":"")
```

Beim Eingeben der Formel in Feld A7 wird die Formel als Matrixformel eingegeben. Das bedeutet, die Eingabe wird mit [Ctrl-Shift-Enter] abgeschlossen. Excel setzt daraufhin die Formel in der Bearbeitungszeile in geschweifte Klammern.

☐ *Ich möchte Tabellen, die mit einem Paßwort geschützt sind, in Excel Version 2.10 D über ein Makro laden. Allerdings werde ich nach Aufruf der Funktion »Laden( )« bei jeder Tabelle nach dem Paßwort gefragt.*

Senden Sie das Paßwort über die Funktion »Tastenf.Senden( )« an Excel. Beachten Sie dabei, daß Tastatureingaben erst dann abgearbeitet werden, wenn Excel zum Beispiel eine Eingabe erwartet. Setzen Sie deshalb »Tastenf.Senden( )« vor den Aufruf von »Laden( )«. Ein Beispiel: Sie haben die Tabelle »Tab1.xls« mit dem Paßwort »Schutz« gesichert

```
=Tastenf.enden("Schutz{eingabe}")
=Laden("tab1.xls")
.....
=Rücksprung( )
```

☐ *Ich habe ein großes Kalkulationsmodell mit mehreren Dateien und umfangreichen Berechnungen entwickelt. Die automatische Neuberechnung ist abgeschaltet. Sobald ich die manuelle Neuberechnung mit [F9] aktiviere, werden immer alle Tabellen durchgerechnet. Kann ich die Neuberechnung auf die aktuelle Tabelle begrenzen?*

Halten Sie die Shift-Taste gedrückt, während Sie das Menü »Optionen« anklicken, beziehungsweise drücken Sie [Alt-Shift-O], wenn Sie mit der Tastatur arbeiten. Das darauf angezeigte Menü gleicht weitgehend dem gewohnten – allerdings können Sie jetzt »Datei berechnen« anstelle von »Neu berechnen« wählen.

☐ *Wie kann ich Daten aus Excel in den Pagemaker übernehmen?*

Da es sich bei beiden Programmen um Windows-Applikationen handelt, kann der Datenaustausch ganz einfach über die Windows-Zwischenablage durchgeführt werden. Es stehen Ihnen mehrere Alternativen offen, wobei unterschieden wird, ob Sie unformatierte Texte und Werte, formatierte Texte und Werte oder Diagramme übernehmen wollen.

Um Texte und Werte als unformatierten Text zu übernehmen, kopieren Sie den gewünschten Bereich in die Zwischenablage (markieren, anschließend »Bearbeiten, Kopieren« oder »Ausschneiden«) und fügen ihn direkt in Pagemaker ein. Haben Sie dabei einen Bereich über mehrere Spalten kopiert, so müssen Sie noch für jede Spalte einen Tabulator setzen.

Wollen Sie formatierten Text und Werte übernehmen, markieren Sie ebenfalls den gewünschten Bereich. Halten Sie die Shift-Taste gedrückt, und wählen Sie »Bearbeiten«. Falls Sie mit der Tastatur arbeiten, drücken Sie [Shift-Alt-B]. Wählen Sie »Bild kopieren« und im darauf ausgegebenen Dialogfeld je nach Bedarf »Wie angezeigt« oder »Wie ausgedruckt«. Das Format »Wie ausgedruckt« liefert, abhängig vom verwendeten Drucker beziehungsweise Druckertreiber, meist bessere Ausdruckergebnisse, erfordert aber auch erheblich mehr freien Speicher. Außerdem können Sie zwischen verschiedenen Formaten wählen. Standardmäßig wird in beiden Fällen ein Vektorformat erzeugt, das (falls Sie die Zwischenablage mit dem Programm »ablage.exe« einsehen) als »Picture« gekennzeichnet wird. Wenn Sie die Shift-Taste gedrückt halten, während Sie auf »OK« klicken, so wird eine Bitmustergrafik (»ablage.exe«: »Bitmap«) kopiert.

Beim Format »Wie ausgedruckt« können Sie ferner bestimmen, ob Zeilen- und Spaltenköpfe und Gitternetzlinien mitkopiert oder ausgeblendet werden sollen. Verwenden Sie dazu den Befehl »Datei, Layout«. Unter Umständen werden mit »ablage.exe« beim Picture-Format, abhängig vom Skalierungsfaktor, nicht alle Linien und Schriftzeichen korrekt angezeigt. Der Ausdruck ist allerdings in jedem Fall korrekt.

Ein Diagramm können Sie wie formatierten Text kopieren. Zusätzlich können Sie die Größe der Kopie zwischen »Wie angezeigt« und »Wie ausgedruckt« auswählen. Auch die Steuerung von Bitmap- und Vektorformat ist gleich.

☐ *Ich arbeite in Excel häufig mit mehreren Dateien. Wie kann ich mehrere Dateien in einem Arbeitsschritt laden und speichern?*

Dazu gibt es die Arbeitsbereichsdateien. Wenn Sie mehrere Tabellen, Diagramme oder Makro-Vorlagen geladen haben, erzeugen Sie eine Arbeitsbereichsdatei mit »Datei, Arbeitsbereich speichern«. Im daraufhin angezeigten Dialogfeld geben Sie einen beliebigen Namen ein. Sofern Sie keine Erweiterung angeben, wird standardmäßig »xlw« vergeben. Anschließend wird jede Datei nach Rückfrage gespeichert und die Arbeitsbereichsdatei angelegt. Diese enthält keine Tabellen, Diagramme, Makros oder ähnliches, sondern nur Informationen über die geladenen Dateien und deren Anordnung auf dem Bildschirm.

Wenn Sie nun zum Beispiel eine solche Datei am Montag herstellen und den Namen »montag« dafür vergeben, könnten Sie am Dienstag die Datei »montag.xlw« laden, und Excel lädt wieder alle Dateien und ordnet sie so an, wie Sie sie am Tag zuvor gespeichert hatten. Um alle Dateien in einem Arbeitsschritt zu schließen, halten Sie die Shift-Taste gedrückt, während Sie das Menü »Datei« anklicken, oder wenn Sie mit der Tastatur arbeiten, drücken Sie [Alt-Shift-D]. Das daraufhin angezeigte Menü gleicht weitgehend dem gewohnten – allerdings können Sie jetzt »Alles schließen« anstelle von »Schließen« wählen.

Multiplan

☐ *Wie erhalte ich bei der Datumsanzeige auch den Tag korrekt ausgeschrieben (zum Beispiel »Mittwoch, den 14. Juni 1989«)?*

Formatieren Sie mit dem Befehl »Format, Felder, Formatcode« das entsprechende Datumsformat wie folgt:

```
tttt",_den"_tt._mmmm_jjjj
```

Geben Sie die Unterstriche als ein Leerzeichen ein. Es gelten die Regeln: Zwei Buchstaben ergeben eine Zahl (zum Beispiel »tt.mm.jj« entspricht »14.06.89«), drei Buchstaben ergeben die ersten drei Buchstaben des Tages beziehungsweise des Monats (»ttt.mmm.jj« ergibt »Mit.Jun.89«), vier Buchstaben ergeben die komplett ausgeschriebene Bezeichnung des Tages beziehungsweise des Monats (»tttt.mmmm.jjjj« zeigt »Mittwoch.Juni.1989«).

☐ *Ich möchte in Multiplan mit der Datenbank alle Kunden heraussuchen, die vor dem 31.12.88 meine Kunden geworden sind. Gebe ich aber in das Suchfeld mit »Text« die Bedingung »<31.12.88« ein, wird kein Kunde gefunden. Ist das Datum kein gültiges Suchkriterium?*

In Multiplan funktioniert eine Suche in der Datenbank auch mit dem Datum. Da Multiplan aber intern das Datum als eine serielle Zahl ablegt, müssen Sie über ein außerhalb des Suchkriterienbereichs liegendes Bezugsfeld gehen. Geben Sie diesem Feld beispielsweise den Namen »(NAME) 'Such_Datum'«. Setzen Sie in dieses Feld das gesuchte Datum normal mit »Text« ein (hier 31.12.88). In das Feld des Suchkriterienbereichs geben Sie dann ein: »<Such_Datum«. Auf die gleiche Art können Sie auch eine Suchbedingung nach Zeiten stellen.

☐ *Ich habe mit der in Multiplan enthaltenen Datenbank eine Und- und Oder-Suchbedingung aufgestellt. Eigenartigerweise werden bei manchen Eingaben alle Sätze ausgewählt. Wie kann ich das vermeiden?*

Eine Und-Verknüpfung wird in einer Zeile abgefragt, das heißt, alle Bedingungen in einer Zeile sind miteinander »und-verknüpft«. Eine Oder-Verknüpfung erfolgt dagegen zwischen zwei Zeilen (oder mehr). Multiplan überprüft zunächst für jede Zeile die Bedingung, das heißt, ist in einer Zeile keine Bedingung, so ist die gesamte Bedingung der Zeile wahr. Wird diese Zeile nun mit einer anderen Zeile »oder-verknüpft«, ist die gesamte Bedingung wahr – es wird jeder Satz gefunden. Wollen Sie nur eine Bedingung eingeben, so ist es deshalb erforderlich, diese Bedingung in der ersten wie auch in der zweiten Zeile anzugeben.

☐ *Wie kann ich unter Multiplan Zahlen mit führenden Nullen (also »0023,56 DM«) darstellen?*

Dieses Format ist kein Standard in Multiplan. Sie können es aber selbst herstellen. Gehen Sie dazu in »Format-Felder«, und markieren Sie den Bereich für diese Formatierung. Mit der Tabulatortaste gehen Sie dann in »Formatcode« und geben »0000,00 DM« ein. Nach der Bestätigung mit der Enter-Taste wird dieses Format in der Formatbibliothek gespeichert und der markierte Bereich entsprechend formatiert. Dieses Format können Sie dann jederzeit wieder über »Format-Felder-Formatcode« aufrufen.

☐ *Ich möchte mit einem Makro überprüfen, ob das aktive Feld eine Zahl ist. Wenn das Ergebnis dieser Abfrage »wahr« ist, soll das Makro fortfahren, ansonsten soll das Makro abgebrochen werden. Wie muß die entsprechende Formel aufgebaut sein?*

Sie können die Abfrage sehr einfach gestalten. Verwenden Sie zum Beispiel folgende Formel:

```
1=Wenn(Istzahl(aktives.Feld());gehezu(Z(1)S;Rücksprung())
```

Diese Formel überprüft, ob das gerade aktive Feld einen Zahlenwert enthält. Wenn die Abfrage erfolgreich verläuft, fährt das Makro in der Bearbeitung fort, ansonsten wird es beendet.

Star Writer PC 3.0

☐ *Wie kann ich aus einer ASCII-Datei, die ich mit einem anderen Textprogramm oder auch mit dem Star Writer geschrieben habe, eine Dokumentdatei machen?*

Um eine ASCII-Datei in eine Dokumentdatei umzuwandeln, gehen Sie folgendermaßen vor:
Laden Sie eine neue Dokumentdatei, und bewegen Sie die Schreibmarke am linken Bildschirmrand etwa auf die Bildschirmmitte. Drücken Sie nun die Tastenkombination [Ctrl-00]. Sie werden nach einem zweiten Fensternamen gefragt. Geben Sie jetzt den Namen der zu konvertierenden ASCII-Datei ein, und beenden Sie den Eintrag mit der Enter-Taste. Markieren Sie den Textanfang in der ASCII-Datei mit der Taste [F7] und das Textende mit der Taste [F8] als Block. Durch einen Druck auf die Taste [F1] gelangen Sie jetzt in das obenliegende Dokumentfenster. Betätigen Sie die Tastenkombination [Ctrl-KC]. Sie haben mit diesen Befehlsfolgen den Inhalt der ASCII-Datei in eine Dokumentdatei importiert. Die Dokumentdatei kann jetzt mit allen Formatierungen, die der Star Writer 3.0 bietet, bearbeitet werden.

☐ *Bei der Eingabe der Option »Einrücken links« im Menü »Absatzlayout« im Zusammenhang mit Blocksatz ergibt sich an der Position für »Einrücken links« bei jedem Absatzanfang eine Verschiebung von ein bis zwei Spalten. Was mache ich falsch?*

Die Option »Einrücken links« bezieht sich – an dieser eingestellten Position – nur auf den Fließtext, nicht aber auf die erste Zeile eines Absatzes. Die meisten Anwender gleichen daher den Abstand bis zur Position für »Einrücken links« mit Leerzeichen aus. Dies ist im Blocksatz jedoch nicht erlaubt, weil bei dieser Formatierung die Leerzeichen grafisch erzeugt werden.
Um die von Ihnen genannte Einrückung zu erhalten, können Sie sich eines kleinen Tricks im Absatzlayout bedienen: Setzen Sie im Menüpunkt »Layout« den »Absatzeinzug« auf einen Wert, der um drei Zeichen kleiner ist als die »Einrückung links«. Danach erzeugen Sie einen Tabulatorsprung – genau auf der gleichen Position wie die »Einrückung links«. Dabei darf der Tabulator in der oberen Linealzeile nicht angezeigt werden, weil er sonst durch das »L« von der Position »Einrückung links« überschrieben wird.
Wenn Sie nun einen neuen Absatz beginnen, drücken Sie (um mit dem Satzanfang auf die Position für »Einrücken links« zu gelangen) einmal die Tabulatortaste. Schreiben Sie dann den Absatz zu Ende. Auf dem Ausdruck werden Sie sehen, daß alles sauber untereinander steht.

☐ *Im Grafikprogramm meines Star Writer 3.0 hinterläßt die Maus nach jeder Bewegung kleine Punkte auf dem Bildschirm. Wie läßt sich hier Abhilfe schaffen?*

Laden Sie das Installationsprogramm des Star Writer 3.0, und wählen Sie hier den Menüpunkt »Grafikprogramm« aus. In diesem Menü wechseln Sie mit der Taste [PgDn] auf die zweite Bildschirmseite. Hier tragen Sie unter dem Punkt »Hardware Besonderheiten« eine 1 ein. Mit [Esc] verlassen Sie das Installationsprogamm. Sichern Sie die Eintragungen, und starten Sie das Grafikprogramm erneut. Die von der Maus erzeugten Pünktchen werden nun nicht mehr auftreten.

David PC

☐ *Trotz erfolgreicher Installation der Datenbank David PC konnte ich den Druckertreiber für meinen Epson-LQ 850 im Programm nicht aktivieren. Wo liegt der Fehler?*

Bei der Installation haben Sie den Treiber vermutlich nicht mit der Leertaste ausgewählt. Eine Nachinstallation können Sie durchführen, indem Sie die Prozedur von Laufwerk A erneut starten. Wählen Sie bitte bei der Abfrage »Erstinstallation« die Box »Nein« und den entsprechenden Druckertreiber jeweils mit der Leertaste aus. Sie sehen jetzt im zweiten Fenster den gewünschten Treiber. Die

Box »OK« bestätigen Sie dann ebenfalls mit der Leertaste. Nachdem Sie die Druckertreiber in das Arbeitsverzeichnis kopiert haben, beantworten Sie die anschließende Frage »Zurück zu DOS« mit einem [Enter].

Star Manager 2.0

☐ *Warum kann ich die speicherresidenten Teile des Star Manager nicht aus einer Anwendung heraus aufrufen?*

Es kann passieren, daß die Tastenbelegung, die im Programm vorgegeben ist, durch andere installierte Programme im Speicher abgefangen wird. Dies ist für den Anwender nicht unbedingt nachvollziehbar, weil die Tastenbelegungen von der Programmierung her auch im Hintergrund stehen können und deshalb gar nicht sichtbar sind. Sie können dies aber umgehen, indem Sie unter dem Menüpunkt »Optionen/Popup-Tasten« eine Tastenumbelegung vornehmen. Sie sollten hierbei jedoch die Verwendung der Alt- und Control-Taste vermeiden. Diese beiden Tasten werden in vielen Kombinationen genutzt und sind meistens die Ursache für eine Blockierung.

☐ *Was bewirkt der Befehl »Platte/Optimieren«?*

Alle Dateien auf einer Festplatte oder Diskette sind der Reihe nach abgelegt. Wenn Dateien gelöscht werden, sind diese zunächst als »gelöscht« gekennzeichnet. Werden jetzt neue Dateien hinzugeschrieben, so prüft das Programm, ob als »gelöscht« gekennzeichnete Dateien vorliegen. Es nutzt dann diesen Platz, um den Datenträger an dieser Stelle wieder aufzufüllen.
Es kann vorkommen, daß besonders große Dateien »weit verstreut« auf dem Speichermedium liegen. Das Ergebnis ist dann eine Verzögerung der Zugriffszeit. Um diesen Effekt zu vermeiden, können Sie die Ordnung auf dem Datenträger optimieren. Das heißt, die Dateien werden so abgelegt, daß sie in zusammenhängenden Clustern oder Sektoren angeordnet sind. Dadurch ist jetzt eine optimale Grundlage für den Zugriff auf die Dateien und deren Bearbeitung gegeben.
Bevor Sie nun eine Optimierung Ihres Datenträgers vornehmen, müssen Sie ihn zunächst prüfen, um eventuelle Fehler zu korrigieren. Es können zum Beispiel defekte Sektoren vorliegen, die bei Nichtbeachtung zu Datenverlusten oder gar Zerstörungen führen. Durch die Neuordnung der Dateien werden die Cluster zusammenhängend auf einem freien Platz im »hinteren« Teil des Speichermediums abgelegt. Um die Dateien zu sortieren, werden zwei Dateien angelegt. In diesen zwei Dateien werden die zu sortierenden Dateien in gleicher Reihenfolge abgelegt und miteinander verglichen. Sollten dabei Unregelmäßigkeiten auftreten, kann das Programm seine Arbeit nicht mehr fortführen, und die Daten sind verloren.

☐ *Warum kann ich unter dem Menüpunkt »Platte/Kopieren« nur die Laufwerke A und B anwählen?*

Damit wird der gesamte Inhalt von einer Diskette auf eine andere kopiert, ohne jede Datei extra aktivieren zu müssen. Durch diesen Befehl wird auch das Format der Quelldiskette auf die Zieldiskette übernommen.
Um diese Funktion ausführen zu können, benötigt der Star Manager die MS-DOS-Datei »diskcopy.com« oder »diskcopy.exe«. Diese muß sich wiederum entweder im aktuellen Verzeichnis oder in einem der Verzeichnisse befinden, die im Pfad in der Datei »autoexec.bat« eingetragen sind.
Es können nur die Laufwerke A und B ausgewählt werden, weil der Inhalt einer Festplatte mit ihren Formatierungen ohnehin nicht auf eine Diskette passen würde. Natürlich können Sie diesen Befehl auch mit einem Diskettenlaufwerk nutzen. Hier werden Sie dann vom Programm aufgefordert, die jeweilige Diskette einzulegen.

Star Planer

☐ *Beim Drucken aus dem Star Planer werden sehr oft unregelmäßige Seitenvorschübe von meinem Drucker durchgeführt. Wo kann der Fehler liegen?*

Zunächst überprüfen Sie bitte die Einstellung der Seitenlänge. Diese muß bei der Verwendung von Endlospapier mit 72 Zeilen angegeben sein. Wenn diese Angabe stimmt, das Problem aber weiterhin auftritt, so haben Sie vergessen, den Befehl »Justieren« einzugeben. Sie müssen vor jedem Betrieb des Druckers nach der korrekten Einstellung des Papiers im Drucker und vor dem Druckbefehl im Star Planer den Befehl »Justieren« bestätigen. Dieser Befehl bewirkt, daß die interne Zeilenzählung des Druckers auf »Null« gesetzt wird und so eine einwandfreie Papierführung gewährleistet ist. Sobald Sie manuell die Papierstellung verändern, müssen Sie diesen Befehl erneut eingeben.

☐ *Verfügt der Star Planer über Datenbankfunktionen, und wie funktionieren diese?*

Sie können in einem Star-Planer-Arbeitsblatt einen Bereich definieren, der als Datenbank genutzt werden kann. Im Menü wählen Sie zu diesem Zweck die Befehlskette »Daten, Abfrage, Bereich«. Jetzt können Sie Daten in diesen Bereich eingeben. Über diesem Datenbereich sollte jedoch immer genügend Platz vorhanden sein, um dort später Suchfunktionen und -kriterien eingeben zu können. Des weiteren können Sie einen Ausgabebereich definieren, in dem die den Suchkriterien entsprechenden Datensätze automatisch angezeigt werden.

☐ *Beim Ausdrucken von Texten aus dem Star Planer 2.0 erfolgt kein Zeilenvorschub. Was habe ich übersehen?*

Bei der Voreinstellung im Menü »Druckereinstellung« haben Sie den Punkt »Auto-ZV« (automatischer Zeilenvorschub) auf »Ja« gesetzt. Dadurch setzt der Star Planer einen automatischen Zeilenvorschub am Drucker voraus und sendet kein Signal für den Zeilenvorschub. Ändern Sie also die Voreinstellung unter »Auto-ZV« auf »Nein« und geben Sie den Befehl »Aktualisieren«, so sendet der Star Planer bei jedem Ausdruck das entsprechende Signal an den Drucker, und ein korrekter Zeilenvorschub wird durchgeführt.

Wordstar 5.0 Professionell

☐ *Bei der Arbeit mit Proportionalschrift kann ich die Makro-Funktion von Wordstar 5.0 Professionell nicht voll nutzen. Wie kann ich dem abhelfen?*

Zur Abhilfe schreiben Sie sich eine Programmdatei mit folgendem Inhalt:

```
IBMFLG-2=40,94
```

Speichern Sie diese Datei unter einem beliebigen Namen, und starten Sie das Programm »wschange«. Geben Sie die Option »E« ein, um in den »Auto-Patcher« zu gelangen. Dort geben Sie »A« ein für »Patch von Diskette einlesen«. Nun geben Sie den Namen der Programmdatei ein, in der Ihr Patch gespeichert ist. Die Einschränkung der Makro-Funktion ist damit aufgehoben.

☐ *Wie kann ich Steuersequenzen, die den Wert Null enthalten, aus Wordstar 5.0 an den Drucker senden?*

Um beispielsweise eine Steuersequenz an den Drucker zu senden, ist es notwendig, das Zeichen 00hex zu übermitteln. Halten Sie die Taste [Alt] gedrückt, und betätigen Sie gleichzeitig die Taste [0] auf der alphanumerischen Tastaur (nicht auf dem Ziffernblock). Jetzt wird das Zeichen 00hex eingefügt.

Betriebssystem und Programmiersprachen

Windows/386

☐ *Ich habe eine QMS-Jetscript-Karte in meinem HP-Laserdrucker, kann jedoch keine Postscript-Datei ausdrucken. Wie muß ich hier vorgehen?*

Wenn Sie über Ihre Windows-Systemsteuerung einen Postscript-Drucker installiert haben und Windows druckt dennoch nicht aus, gehen Sie bitte wie folgt vor:

1.) Ändern Sie den Device-Eintrag für Ihre QMS-Jetscript in »-IO«.
2.) Entfernen Sie den Jumper »JW1« auf der Controller-Karte.
3.) Erzeugen Sie eine Datei mit folgendem Inhalt:

```
serverdict begin 0 exitserver
statusdict begin
1 setdefaultparallel
1 setparallel
start
^d
^c
```

4.) Kopieren Sie diese Datei zum Drucker (Copy <Dateiname> LPT<Nummer>).
5.) Dieser Befehl generiert eine Jetscript-Startup-Page. Falls nicht, geben Sie den Befehl »copy con lpt2« ein, um sicher zu sein, daß der Befehl angenommen wurde. Diesen Befehl müssen Sie nur einmal eingeben, er muß nicht in Ihrer »autoexec.bat« stehen. Das Schließen des »Reverse Channel« hat zwei Effekte: Der Jetscript-Status-Report zeigt keinen Status mehr an, und Sie können keine Adobe-Soft-Postscript-Fonts mehr benutzen.
6.) Um den unter Punkt 3 beschriebenen Befehl wieder rückgängig zu machen, müssen Sie folgende Anweisung an Ihren Drucker senden:

```
serverdict begin 0 exitserver
statusdict begin
1 setdefaultparallel
1 setparallel
start
^d
^c
```

Windows/286

☐ *Ich habe einen Postscript-Drucker, der bei jedem Einschalten eine Startseite ausdruckt. Kann ich dies unterdrücken?*

Schreiben Sie dazu das folgende kleine Programm (das Sie zum Beispiel »nostart.txt« nennen) mit einem Editor, der keine Steuerzeichen einfügt (wie Windows-Notepad oder Windows-Write), und speichern Sie es anschließend über die Funktion »Speichern unter« mit der Option »Nur Text«. Bitte beachten Sie die Schreibweise und die Groß-/Kleinschreibung.

```
serverdict begin 0 exitserver
statusdict begin
false setdostartpage
end
```

Kopieren Sie diese Datei nur einmal auf den Drucker. Beispiel:

```
copy nostart.txt lpt1:
```

oder, falls der Drucker seriell angesteuert wird:

```
mode com1:,96,n,8,1
copy nostart.txt com1:
```

Die Einstellung wird im nichtflüchtigen Speicher (NVR oder ähnliche Bezeichnung) des Druckers abgelegt und bleibt solange erhalten, bis sie wieder geändert wird. Um die Startseite wieder zu aktivieren, schreiben Sie folgendes Programm und verfahren wie oben beschrieben:

```
serverdict begin 0 exitserver
statusdict begin
true setdostartpage
end
```

Falls Sie mit dem IBM-4216-Personal-Pageprinter arbeiten, fügen Sie in Ihrer »autoexec.bat« den Buchstaben »n« in der Zeile »ppstart« hinzu:

```
ppstart /a /nt /l /p
```

C 5.1

☐ *In einem Programm, in dem mehrfach Speicher mittels »malloc( )« zugewiesen und mittels »free( )« wieder freigegeben wurde, schlägt ein Aufruf der Funktion »spawn( )« fehl. Was ist die Ursache?*

Die speicherzuweisenden Funktionen fordern den benötigten Speicherplatz in Blöcken von je 8 KByte vom Betriebssystem an und betreiben innerhalb dieser Speicherblöcke ein eigenes Speichermanagement. Wird Speicher durch die Funktion »free( )« freigegeben, so wird der betreffende Speicherbereich als frei verfügbar gekennzeichnet, jedoch nicht an das Betriebssystem zurückgegeben. Werden nacheinander immer wieder Speicherbereiche unterschiedlicher Größe angefordert und freigegeben, kann es unter ungünstigen Umständen vorkommen, daß angeforderter Speicherbereich aufgrund seiner Größe nicht durch den freigewordenen Speicherplatz abgedeckt werden kann. In diesem Fall wird wieder ein neuer Speicherblock vom Betriebssystem angefordert. Somit kann es passieren, daß das Betriebssystem nicht mehr genügend Speicher für einen Spawn-Prozeß zur Verfügung stellen kann.

☐ *Beim Linken unter dem neuen Betriebssystem OS/2 wird die Fehlermeldung »memory protection violation« ausgegeben und danach das System gestoppt. Was kann ich dagegen unternehmen?*

Der Linker versucht gemäß Voreinstellung, gefundene freie Bereiche in Objektdateien zu kürzen, also den Code zu packen. Hierzu ist es notwendig, Programmteile in den Speicher auszulagern, zu packen und wieder einzubinden. Der Fehler tritt auf, da OS/2 den gesamten Hauptspeicher überwacht und keine unangemeldete Speicherbenutzung zuläßt. Als Lösung ist die Option »/nop[ackcode]« anzugeben, die ein Packen des Codes unterbindet.

☐ *Wenn ich versuche, eine Datei aus dem aktuellen Verzeichnis heraus über einen absoluten Pfad in ein anderes Verzeichnis zu kopieren, so mißlingt dies, wenn das aktuelle Verzeichnis im Pfad »DOS-Append« angegeben wurde. Was ist die Ursache?*

Der mittels »append« spezifizierte Pfad hat absolute Priorität vor allen absoluten Pfadangaben, die von den I/O-Funktionen der C-Bibliotheken benutzt werden. Dies bedeutet, daß der Append-Pfad benutzt wird, ohne Rücksicht darauf, ob ein absoluter Pfad benutzt wurde oder nicht. Abhilfe schaffen Sie, indem Sie entweder das aktuelle Verzeichnis aus dem Append-Pfad nehmen, oder indem Sie kurzfristig mittels »chdir( )« in ein Verzeichnis wechseln, das nicht im DOS-Append-Pfad eingetragen ist.

Editor M 1.0

☐ *Wenn ich den M-Editor unter OS/2 starte, wird die Datei »tools.ini« ignoriert, obwohl die OS/2-Environment-Variable »set init = <Pfad>« richtig gesetzt und die Datei »tools.ini« auch in diesem Verzeichnis vorhanden ist. Was ist die Ursache?*

In der Datei »tools.ini« muß in der ersten Zeile die Direktive »[mep]« gesetzt werden, die anzeigt, daß sich die zugehörige Datei »tools.ini« auf den M-Editor im Protected Mode bezieht.

☐ *Auf einigen Rechnern gelingt es nicht, im Microsoft-Editor das Paragraphenzeichen auf dem Bildschirm auszugeben. Es erscheint beim Drücken von [Shift-3] statt dessen immer die Meldung »CTRL + U is not assigned to any editor function«. Wie kann ich dennoch mit dem Paragraphenzeichen arbeiten?*

Fügen Sie mit dem Editor Edlin, der mit dem Betriebssystem MS-DOS mitgeliefert wird, folgende Zeilen in die Initialisierungsdatei »tools.ini« des Microsoft-Editors ein:

```
para := "§"
para : Ctrl+U
```

und stellen Sie sicher, daß die MS-DOS-Umgebungsvariable »init« mittels des Befehls:

```
set init=<Lw:\><Pfad>
```

auf den Pfad zeigt, in dem sich die Datei »tools.ini« befindet. Wenn Sie jetzt den Microsoft-Editor aufrufen, können Sie das Paragraphenzeichen direkt über die Tastatur ansprechen.

Pascal 4.0

☐ *Kann man auch bei Line-Sequential-Dateien, ähnlich wie bei Random-Access-Dateien, Daten an eine bereits bestehende Datei anhängen?*

Prinzipiell können Sie unter Pascal keine Daten an eine Line-Sequential-Datei anhängen. Sie können jedoch mit dem Befehl »rewrite« eine neue Datei öffnen, die alte Datei mit »reset« öffnen, den Inhalt der alten Datei in die neue Datei einlesen und danach die anzufügenden Daten in die neu geöffnete Datei schreiben.

Quick Basic

☐ *Wie schalte ich die Taste [Caps Lock] per Programm ein und aus?*

Sie können den Status der Tastatur wie folgt lesen:

```
def seg = 0
x = peek(1047) and 64
Falls Caps Lock = ein ist x = 0
Falls Caps Lock = aus ist x = 64
```

(siehe Tabelle). Um die Caps-Lock-Taste auszuschalten, benutzen Sie die Anweisungen:

```
def seg = 0
poke 1047, peek(1047) and 191
```

und die Anweisungen

```
def seg = 0
poke 1047, peek (1047) or 64
```

Zusätzliche Informationen: Jedes Bit der Speicheradresse 1047 steht für den Status eines Tastatur-Flags. Diese enthalten »Num Lock«, »Scroll Lock«, »Caps Lock«, »Einfüge-Modus« und Informationen, ob die linke oder die rechte Shift-, die Alt- oder die Control-Taste gerade gedrückt ist oder nicht.

Um den Wert eines Statusbits zu ermitteln, gilt die Anweisung

```
peek(1047) and <bval>
```

wobei »<bval>« dem dezimalen Wert der abzufragenden Taste entspricht. Der Rückgabewert der Anweisung ist »0«, falls die Taste nicht aktiv, und »1«, falls die Taste aktiv ist.

Um nun ein Statusbit zu setzen (nur gültig für die Lock-Tasten und den Einfüge-Modus), muß dem entspechenden Bit der Wert 1 zugewiesen werden. Dieses wird erreicht durch:

```
poke 1047, peek(1047) or <bval>
```

Ausgeschaltet wird eine Taste durch:

```
poke 1047, peek(1047) and (255 - <bval>)
```

Dabei sollte bedacht werden, daß einfaches Schreiben eines Bytes, dessen relevantes Bit gesetzt ist, die Werte der anderen Statusbits auf »0« setzen würde. Um dieses zu verhindern, sollte unbedingt die oben angesprochene Strategie verfolgt werden.

Bit Nr.	Dezimalwert	Tastatur-Flag
0	1	rechts Shift
1	2	links Shift
2	4	Control
3	8	Alt
4	16	Scroll Lock
5	32	Num Lock
6	64	Caps Lock
7	128	Einfüge-Modus

☐ *Wie kann ich verhindern, daß sich der PC aufhängt, wenn ich mit »qb.exe« den Einzelschrittmodus und »breakpoints« verwende?*

Wird folgendes Programm mit »qb.exe« Version 4.00 bearbeitet (mit »trace on« und »breakpoint« in der Zeile nach der Anweisung »'next i'«), erscheint eine unsinnige Ausgabe, wenn der »breakpoint« erreicht wird. Die OK-Box wird angezeigt, und der Computer hängt. Das Problem wurde bei der Version 4.00b beseitigt.
Dazu ein Programmbeispiel:

```
type testtype
test1 as string * 30
test2 as string * 30
test3 as string * 30
test4 as integer
test1 as integer
end type

dim player(0 to 3) as testtype
for i = 1 to 3
line input "player: ",player(i).test1
next i
```

☐ *Wie fordere ich Speicher für Felder größer als 128 KByte an? Ich bekomme immer eine Fehlermeldung.*

Wenn Datenstrukturen (Arrays, Type) angelegt werden sollen, die größer als 128 KByte sind, ist es erforderlich, die Variablen in ihrer Größe so festzulegen, daß keine einzelne Variable eine Segmentgrenze kreuzt.

Eine Variable kreuzt unter Umständen eine Segmentgrenze, wenn die gewählte Größe keine Potenz von 2 ist. Das untere Beispiel zeigt einen solchen Fall:

```
dim feld(2000) as string*80
2000 * 80 / 1024 = 156,25k Byte
```

Hier würde nun die Variable »Feld(1639)« mit ihren ersten 32 Bytes im zweiten Segment und die restlichen 48 im dritten Segment liegen (ab der Variablen »Felder« ist eine Größe über 128 KByte nicht erlaubt). Dimensioniert man das Feld nun mit:

```
dim feld(2000) as string*128
2000 * 128 / 1024 = 250k Byte
```

wobei »Feld(1024)« die letzte Variable im zweiten 64-KByte-Segment und »Feld(1025)« die erste Variable im dritten Segment ist, dann können Sie mit dieser Methode den gesamten zur Verfügung stehenden Speicher nutzen.

☐ *Warum können keine Dateien mit deutschen Umlauten erzeugt werden?*

Es handelt sich hierbei um ein Problem im Compiler. Ab der Version 4.00b können jedoch alle Umlaute in Dateinamen benutzt werden. Gern erhalten Sie ein kostenloses Update.

☐ *Wenn ich in einem Unterprogramm ein Feld als »static« deklariere, bekomme ich die Fehlermeldung »doppelte Definition«, sobald ich das Unterprogramm ein zweites Mal aufrufe.*

Im Handbuch »Programmieren in Quick Basic, Ausgewählte Themen« ist nicht ausdrücklich erklärt, was passiert, wenn ein Unterprogramm mit einem Static-Feld zum zweiten Mal aufgerufen wird. Das nachfolgende Programm umgeht einen zweiten Aufruf der Dim-Anweisung. Es wird dazu eine Variable benötigt, welche festhält, ob das Programm zum ersten Mal aufgerufen wird, und nur dann die Dim-Amweisung zur Ausführung bringt.

```
declare sub dummy( )
print "call the subroutine"
call dummy
print "call the routine again"

sub dummy( )
 static a( ), firstpassflag
 if firstpassflag = 0 then
    dim a(1)
     firstpassflag = 1
 end if
print "continuing on"
end sub
```

☐ *Kann die Hercules-Karte über das BIOS angesprochen werden, oder kann ich den Hercules-Treiber aus meinem Programm heraus starten?*

Die Hercules-Grafikkarte ist kein Bestandteil des IBM-Standards. Es gibt daher keine Möglichkeit, sie über das BIOS anzusprechen. Die Hercules-Karte ist aber relativ einfach zu programmieren. Der Original-Hercules-Karte liegen entsprechende Bedienungsanleitungen bei, welche die Programmierung dieser Grafikkarte detailliert erklären. Bei Hercules-kompatiblen Karten ist diese Information allerdings oftmals nicht enthalten.
Der Hercules-Treiber »msherc.com« (früher »gbherc.com«) sollte aus laufenden Programmen heraus nicht gestartet werden, da er ein TSR-Programm (»Transient but Stay Resident«) ist, das Hauptspeicherbereiche für sich reserviert. Da dem laufenden Programm – das schon im Startup-Code den Hauptspeicher installiert – diese TSR-Routine dann nicht bekannt ist, kann dies zu Auswirkungen wie Fragmentierung des Hauptspeichers, Verlust von Daten sowie Überschreiben von Programmteilen und somit zum Absturz des PC führen.

Quick C

☐ *Welche Ursachen hat die Fehlermeldung »Load of User Program Failed«?*

Diese Fehlermeldung kann mehrere Ursachen haben:

1.) Es ist nicht genügend Speicherplatz zum Laden des Programms verfügbar. Lösungsvorschlag: entfällt.
2.) Es sind speicherresidente Programme geladen, die in Interferenz mit Quick C diese Fehlermeldung hervorrufen. Lösungsvorschlag: Deinstallieren der TSR-Programme.
3.) Es wurde versucht, eine Exe-Datei im Quick-C-Environment zu starten. Lösungsvorschlag: Löschen aller Exe-Dateien gleichen Namens, die mit »qcl« oder mit »cl« erzeugt wurden.
4.) Quick C konnte die auf der Kommandozeile spezifizierte Bibliothek nicht finden. Lösungsvorschlag: Setzen einer MS-DOS-Environment-Variablen: »set lib = <Pfad zum Bibliotheks-Verzeichnis>«.
5.) Die spezifizierte Quick-Bibliothek ist inkonsistent oder besitzt ein unlesbares Format. Lösungsvorschlag: Erneutes Herstellen der Quick-Bibliothek.
6.) Die Debug-Information der Datei ist inkonsistent oder anderweitig unlesbar. Lösungsvorschlag: erneutes Compilieren mit eingeschalteter Debug-Option.
7.) Die MS-DOS-Environment-Variable »TMP« wurde nicht oder nicht richtig gesetzt. Lösungsvorschlag: Setzen der TMP-Variablen auf ein existierendes Verzeichnis für temporäre Dateien. Dabei ist darauf zu achten, daß vor und hinter dem Gleichheitszeichen keine Leerzeichen eingefügt sein dürfen:

   ```
   set tmp=<Pfad zum Verzeichnis für temporäre Dateien>
   ```

8.) Eine statische Zuweisung von Far-Daten innerhalb des Environment ist nicht erlaubt und sollte die Fehlermeldung C2176 verursachen. Zum Beispiel »long far color[16]«

Vereinzelt wird statt dessen die Fehlermeldung »Load of User Program Failed« ausgegeben. Lösungsvorschlag: Änderung der entsprechenden Programmzeilen und erneute Programmerzeugung mittels »rebuild all«

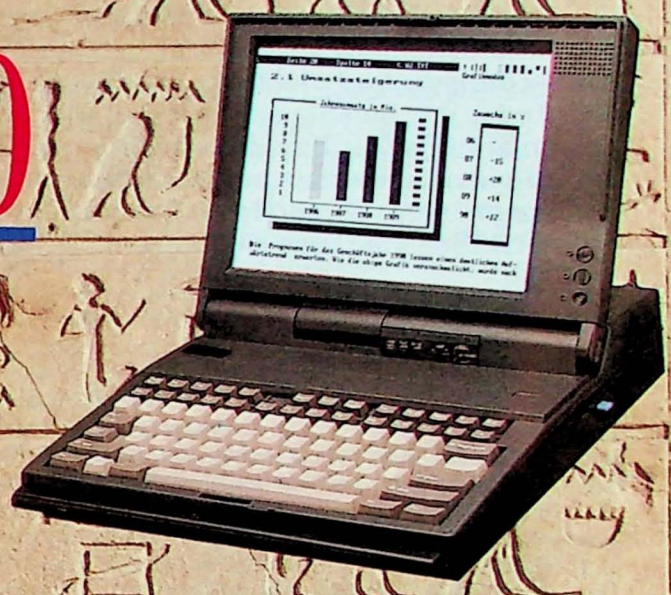
☐ *Ich möchte unter Quick C 2.00 mehr als 20 Dateien öffnen. Obwohl ich die Variable »files« in der Datei »config.sys« bereits auf den entsprechenden Wert geändert habe, ist es mir nicht gelungen, mehr als 15 Dateien gleichzeitig zu öffnen. Was ist die Ursache hierfür, und wie kann ich das Problem umgehen?*

Das Betriebssystem MS-DOS bietet ab der Version 3.30 standardmäßig die Möglichkeit an, mehr als 20 Dateien gleichzeitig zu öffnen. Von diesen 20 Dateien muß jedoch die Anzahl der bereits geöffneten Standardgeräteeinheiten abgezogen werden. Dies sind »STDIN«, »STDOUT«, »STDERR«, »STDAUX« und »STDPRN«. Demnach verbleiben unter normalen Umständen nur mehr 15 Dateien, die gleichzeitig geöffnet werden können.
Für jede Datei wird (zur Verwaltung durch das Betriebssystem) Speicherplatz für die Ablage von Dateiinformationen benötigt. Standardmäßig werden diese Informationen im Programm-Segment-Präfix (PSP) der entsprechenden Exe-Datei abgelegt, welcher den Kopf einer ausführbaren Datei bildet. Da hier jedoch nur Speicherplatz für die Verwaltungsinformation von 20 Dateien bereitsteht, muß dieser, sollen mehr als 20 Dateien geöffnet werden, zusätzlich vom Betriebssystem angefordert werden.
Da Programme jeder beliebigen Größe unter MS-DOS sofort nach Ihrem Aufruf den gesamten Speicherbereich des Betriebssystems für sich beanspruchen, können Sie nicht ohne weiteres über den Interrupt 21hex, Fn 67hex (Set File Handle Count) die Anzahl der zu öffnenden Dateien erhöhen, weil dadurch implizit weiterer Speicherbereich für die Anlage von Dateiverwaltungsinformation vom Betriebssystem angefordert wird. Somit muß zuerst aller Speicherbereich an das Betriebssystem zurückgegeben werden, der nicht vom Programm selbst benötigt wird. Dazu ist es nötig, die Länge des Programms im Speicher (die sich durchaus von der Länge des Programms auf der Festplatte oder Diskette unterscheiden kann) zu ermitteln und über den Interrupt 21hex, Funktion 4Ahex den reservierten Speicherplatz an die Programmlänge anzupassen. Jetzt erst kann mittels des Interrupt 21hex, Funktion 67hex die Anzahl der maximal zu öffnenden Dateien neu eingestellt werden. Zu beachten ist hierbei, daß zum einen der neu eingestellte Wert nur für das Programm gilt, welches diesen Wert neu eingestellt hat, und zum anderen dieser Wert auch in der »config.sys« mit der Variablen »files« eingestellt sein muß, welche die systemweit erlaubte Anzahl an geöffneten Dateien bestimmt.

(as)

Textverarbeitung im Wandel der Zeit

ConText PRO



Grabdenkmal im Besitz des Kestner-Museums, Hannover

Textverarbeitung – ein Thema so alt wie die Kulturgeschichte der Menschheit: was einst in Stein gehauen wurde, ist heute zu einer Aufgabe für den Computer geworden.

Je besser die Computer werden, um so mehr kommt es auf eine leistungsfähige Software an. Um gute Dokumente zu erstellen, brauchen Sie ein Programm, das Ihnen jeden Wunsch von den Tasten abliest und schnell und einfach erledigt: ConText PRO.

Von ConText PRO werden Sie nicht nur auf Super-PCs, sondern auch im Diskettenbetrieb ohne Festplatte oder auf Reisen begeistert sein, denn er ist drei- bis viermal kleiner als vergleichbare Programme und dadurch extrem handlich und schnell.

Neben den Standardfunktionen, die Sie zur Textverarbeitung brauchen, bietet ConText PRO Ihnen noch eine Reihe von besonders durchdachten Extras:

- IBM-Blockgrafik-Editor mit Radiergummi und Eckenautomatik
- Mehrfachausdruck und Drucken, während Sie weiterarbeiten
- Die Undo-Funktion macht Ihre Fehlbedienung ungeschehen
- Deutsche Trennhilfe mit 99prozentiger Treffsicherheit
- Zeit, Datum, Kalender, Sonderzeichen mit Übernahmefunktion
- Suchen und Ersetzen auch mit Sonderzeichen oder Jokern
- Fünf unabhängige Textbereiche mit Übernahmefunktionen
- Schnelles Clipboard (Zwischenablage) für Textabschnitte
- Korrekturtaste für "Verdreher" und Groß-/Kleinschreibung
- Alle Blockfunktionen und Rechnen im Text auch spaltenweise
- DOS-Aufrufe von anderen Programmen aus ConText PRO heraus
- Rechnen von Summe und Durchschnitt für Zeilen und Spalten
- Auf Wunsch zweispaltiges Layout für Druck und Screen
- Kopf- und Fußzeilen mit Text, Seitennummern und Layout
- Full-Page-Anzeige mit bis zu 60 Zeilen auf dem Bildschirm
- Alle Tasten mit Makros oder ganzen Textblöcken belegbar

ConText EASY

Jetzt NEU!

"Einschalten und loslegen"!

ConText EASY ist eine speziell für den Einsteiger oder Hobbyanwender zugeschnittene Textverarbeitung, die alle Vorzüge von ConText PRO mit einfachem und praxisgerechtem Bedienungskomfort verbindet.
Vergessen Sie das Wälzen von Handbüchern und Anleitungen:
ConText EASY heißt "Einschalten und loslegen"! Ohne große Vorkenntnisse oder Computererfahrung kann jeder mit ConText EASY ansprechende Ergebnisse erzielen.

ConText PRO, das Textprogramm für den professionellen Einsatz	DM 199,–*
ConText EASY, "Einschalten und loslegen"	DM 99,–*
ConText ADREVA, universelle Adreßverwaltung	DM 49,–*
Update auf ConText PRO	DM 129,–*
Demonstrationsdiskette	DM 5,–
DMV-Power-Pack ERGO 3.0, ConText PRO, ConText ADREVA	DM 349,–*

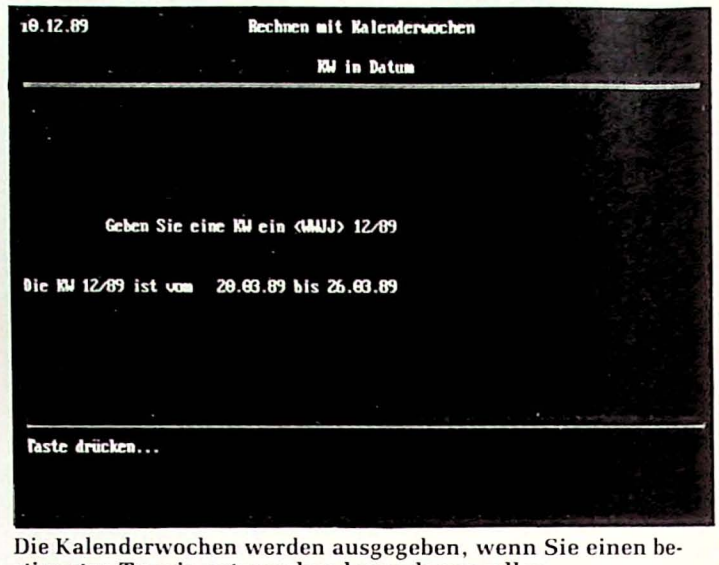
* Unverbindliche Preisempfehlung. Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Tips & Tricks zu dBase III Plus

Eine so umfangreiche Programmiersprache wie dBase III Plus, mit der so viele Anwendungen konzipiert werden, lebt nur von vielen Tips und der umfassenden Kenntnis ihrer Befehle. Die Anwendungen sind nicht nur auf die korrekte Benutzung der Befehle angewiesen, sondern auch auf ein erträgliches Laufzeitverhalten, weil zuweilen viele Datensätze in die Ausführung miteinbezogen sind. Diese Tips & Tricks-Ecke zu dBase III Plus kümmert sich darum.



Die Kalenderwochen werden ausgegeben, wenn Sie einen bestimmten Termin entsprechend zuordnen wollen

dBase mit Filter

Der Sinn einer jeden Datenbank besteht darin, durch sie Informationen zu gewinnen. Da meistens nicht nach dem gesamten Datenbankinhalt gefragt ist, muß eine Datenbankabfragesprache Befehle beinhalten, mit denen der Anwender schnell und gezielt eine einzelne Information erhält. dBase unterstützt vielfältige Vorgehensweisen, um gezielt Daten abzufragen. Dabei denkt der dBase-Anwender zuerst einmal an die zahlreichen Suchbefehle wie »locate« und »find« und an die Datenausgabebefehle mit Bereichsbeschränkung wie zum Beispiel »display ... for«. Weniger bekannt, aber schnell, flexibel und komfortabel ist die Filtertechnik, die dBase III Plus diesen Befehlen gegenüberstellt.

Repräsentiert durch die Befehle »set filter to« und »create query« vereinfacht diese Technik die gezielte Informationssuche und erlaubt Ihnen sogar, einen Bereich einer Datei zur Bearbeitung abzutrennen, ohne dazu erst aufwendig Dateiteile hin- und herzukopieren.

Am einfachsten läßt sich die Arbeitsweise von »set filter« anhand eines Beispiels erklären. Angenommen, Ihr System beinhaltet eine Rechnungsdatei, und Sie möchten sich über alle Rechnungen des Kunden »Muster« informieren. Zunächst wählen Sie mit

```
use rech
```

eine Datenbank. Als nächstes definieren Sie den benötigten Filter. Das sieht so aus:

```
set filter to name = "Muster"
```

Wenn Sie nun

```
list
```

eingeben, listet dBase alle Datensätze auf, die der Filterbestimmung entsprechen. Mit Befehlen wie

```
display all name,rechdat,bemerkung
```

läßt sich die Datenbankdarstellung je nach Anwendung auf die interessanten Informationen beschränken. Dank »set filter to« betrachtet dBase nur noch den von Ihnen als Filterbedingung definierten Bereich.

Wenn die Rechnungsliste, die Ihnen dBase jetzt ausgegeben hat, immer noch zu unübersichtlich ist und Sie nach einer bestimmten Rechnung fahnden, schränken Sie die Filterbedingung einfach weiter ein. Bezogen auf das Beispiel könnten Sie dBase auch dazu bringen, nur alle Rechnungen des Kunden »Muster« ab einer bestimmten Betragsgrenze (hier: 1000 Mark) zu berücksichtigen. Der Befehl dafür lautet (wenn das Feld mit der Bezeichnung »rechbetr« den Rechnungsbetrag enthält):

```
set filter to name = »Muster« .and. rechbetr > 1000
```

Bei den Bereichseinschränkungen sind Ihnen keine Grenzen gesetzt: Wenn dBase nach dieser Filterdefinition zu wenige oder immer noch zu viele Datensätze anzeigt, spannen Sie das Netz einfach durch weitere Filterbedingungen (zum Beispiel »Rechnungsdatum«) enger oder erweitern es wieder, indem Sie zum Beispiel die Betragsgrenze auf 500 Mark herabsetzen.

Als Verknüpfungen sind ».and.«, ».or.«, ».and. .not.« und ».or. .not.« erlaubt. Als Vergleichsoperand ist neben den bisher verwendeten Operanden unter anderem auch das Dollarzeichen »$« zulässig. Seine Funktion läßt sich am besten durch »soll enthalten« ausdrücken. Ein einfaches Beispiel für seine Anwendung ergibt sich, wenn eine Datei ein alphanumerisches Datenfeld enthält, das aus der Postleitzahl und der Ortsangabe besteht, und Sie nur nach dem Ort suchen möchten. Der Filterbefehl

```
set filter to "München" $ plzort
```

erfüllt diesen Zweck. Zurück zur Filterbedingung

```
set filter to name = "Muster"
```

Wenn Sie nun die Gesamtzahl aller Rechnungen wissen möchten, geben Sie einfach

```
count
```

ein. Das verdeutlicht, daß sich alle Befehle, die den Dateiinhalt betreffen, an die von »set filter« vorgegebene Bereichseinschränkung halten. Auch

```
sum(rechbetr)
```

summiert wirklich nur die Rechnungsbeträge der Rechnungen an Herrn »Muster«.

Ein Filter tritt immer erst dann in Kraft, wenn der Datensatzzeiger nach der Filterdefinition das erste Mal bewegt wurde. Nach der Eingabe der Filterdefinition zeigt

```
display
```

immer noch den zuletzt aktuellen Datensatz an. »skip« findet daraufhin aber nur noch Sätze, die dem Filter entsprechen. Der Filterbedingung entsprechen auch die Ausgaben der Befehle »display all« und »list« – selbst wenn Sie sie gleich nach der Filterdefinition eingeben –, weil dBase den Datensatzzeiger zur Ausführung dieser Befehle automatisch an den Dateianfang setzt.

Filter bauen nicht aufeinander auf, sondern es gilt grundsätzlich nur der zuletzt eingegebene Filter: Die Eingabe

```
set filter to name = "Muster"
```

und nachfolgend

```
set filter to rechbetr > 750
```

bewirkt folglich nicht das gleiche wie

```
set filter to name = "Muster" .and. rechbetr > 750
```

Ausschlaggebend ist der letzte Filter. dBase berücksichtigt im ersten Fall also alle Rechnungen mit einem Betrag von mehr als 750 Mark. Im zweiten Fall werden folglich nur die Rechnungsbeträge des Herrn Muster auf 750 Mark abgeprüft.
Ein Filter gilt immer nur für den aktuellen Datenbankbereich. Wenn Sie also mehrere Dateien im Speicher haben, können Sie für diese verschiedene Filterbedingungen definieren:

```
set filter to
```

beendet die Wirkung des Filters, und der gesamte Datenbankinhalt steht wieder zur Abfrage zur Verfügung. dBase gestattet es Ihnen auch, bereits vorher gespeicherte Filterdefinitionen zu verwenden. Dies geschieht mit

```
set filter to datei
```

Um eine Filterdefinition zu speichern, bedienen Sie sich des Befehls »create query«, um sie zu ändern, ziehen Sie »modify query« heran. Nach dem Aufruf erscheint ein bildschirmorientierter »Abfragegenerator«, der im Prinzip nicht viel mehr tut, als eine Filterdefinition festzulegen. Zuerst wählen Sie die Pulldown-Maske »Setze Filter« und tragen einen Feldnamen ein, beziehungsweise wählen einen aus der angezeigten Feldliste aus. Danach lassen Sie sich durch die Betätigung von [Enter] die Liste der Operanden ausgeben und entscheiden sich für einen. Hinter »Konstante/Ausdruck« tragen Sie sozusagen »die rechte Hälfte des Vergleichs ein«. Möchten Sie

```
set filter to name = "Muster"
```

eingeben, tragen Sie demnach »name« als Feldname, »Ist gleich« als Operand und »"Muster"« als Ausdruck ein. Achten Sie bei der Eingabe von Konstante und Ausdruck auf jeden Fall auf die korrekte Anwendung von Anführungszeichen. Wenn Sie eine numerische Variable als Feldname gewählt haben, sind Anführungszeichen hier natürlich fehl am Platze, bei alphanumerischen Variablen dürfen Sie allerdings nicht vergessen werden.
Nach der Anwahl von »Anzeige« listet Ihnen dBase alle zutreffenden Datensätze auf. Das Pulldown-Menü »Ende« bietet Ihnen an, Ihre Suchanfrage auf Diskette oder Festplatte zu speichern. Die von diesem Menüpunkt abgespeicherten Qry-Dateien laden Sie wieder in den Speicher, indem Sie die bereits genannte Anweisung

```
set filter to datei
```

eingeben. Wie unter dBase üblich, haben Sie die Möglichkeit, die mit »create« entworfenen Dateien mit dem entsprechenden Modify-Befehl zu ändern. In diesem Falle verwenden Sie »modify query«.
Bisher haben Sie die Filter immer nur auf der dBase-Kommandoebene eingesetzt. Auch innerhalb von Programmen funktioniert die Anwendung von »set filter« und die Einbindung von mit »create query« definierten Filtern nach dem gleichen Muster. Aufgrund der unkomplizierten Handhabung des Set-Filter-Befehls ist es für Programmierer kein Problem, den Anwendern seiner Programme die Definition von eigenen Filtern anzubieten, indem der Filterbefehl einfach im Zusammenhang mit Variablen eingesetzt wird, die vom Anwender abgefragt werden.
Sehr anwenderfreundlich macht sich beispielsweise eine Abfrage mit folgendem Text: »Bitte geben Sie ein, bei welchem Betrag die Rechnungen berücksichtigt werden sollen (z.B. > 500, < = 2000, etc.)«. Angenommen, Sie verwenden für diese Abfrage die Variable »grenze«, dann sieht die Realisation von »set filter to« folgendermaßen aus:

```
set filter to rechbetr &grenze
```

Sie haben also die Variable »grenze« durch das vorangestellte kaufmännische Und-Zeichen als Makro benutzt.
Es ist bekannt, daß unter den dBase-III-Plus-Versionen 1.0 und 1.1 im Zusammenhang mit »set filter to« ein Fehler auftritt. Die Fehlermeldung (Nummer 67) lautet »Internal Error: EVAL work area overflow« beziehungsweise »Interner Fehler: EVAL Arbeitsbereich-Überlauf«. Laut dBase-Handbuch bedeutet dieser Fehler, daß »Sie versuchen, einen Ausdruck zu verwenden, der zu komplex und nicht möglich ist«. Der Fehler tritt allerdings nur dann auf, wenn Sie lokale (»private«) Variablen in der Filterbedingung verwenden und diese mit ».and.«, ».or.« oder ».not.« verknüpfen. Als Ausweg bleibt Ihnen nur, kritische Variablen mit dem Public-Befehl als global zu kennzeichnen.

(Ulrich Spranger/al)

Diskettenverzeichnisse einlesen

Jeder dBase-Programmierer kennt das Problem: Ein Diskettenverzeichnis muß eingelesen werden. Das Programm »ausdruck.prg« (Listing 1) zeigt, wie es geht. Das Programm ist allerdings nicht nur ein einfaches Beispiel, sondern bringt tatsächlich einigen Nutzen. »ausdruck.prg« hilft Ihnen, wenn Sie mehrere Dateien ausdrucken möchten – zum Beispiel alle Programmdateien, die zu einem Programmsystem gehören, oder aber verschiedene Textdateien, die beispielsweise nicht alle mit dem gleichen Anfangsbuchstaben beginnen und mit vielen anderen Textdateien zusammen in einem Unterverzeichnis stehen, also nicht mit einem einzigen Befehl per »Copy ... prn« auszudrucken sind. Mit »ausdruck.prg« wählen Sie aus dem Inhaltsverzeichnis einfach die entsprechenden Dateien aus und »lassen« drucken.
Das Programm »ausdruck.prg« fragt zuerst nach dem Dateimuster. Damit ist ein Dateiname gemeint, der auch Sternchen und Fragezeichen (die sogenannten »wildcards«) enthalten kann. Diese Variable namens »muster« verwendet das Programm im folgenden MS-DOS-Befehl, der – wie immer unter dBase – mit einem vorangestellten Ausrufezeichen gekennzeichnet ist. Dieser MS-DOS-Befehl speichert das Inhaltsverzeichnis in der Datei »dateien.tmp« – und zwar im ASCII-Format (»muster« ist hier als Makro für die Dateienbeschreibung eingesetzt). Wenn der Benutzer bei der Eingabe des Dateimusters einfach [Enter] drückt, liest der Dir-Befehl das gesamte Verzeichnis. Um diese Datei nun für dBase nutzbar zu machen, muß zunächst einmal eine Datei im dBase-Format angelegt werden, die hier den Namen »dateien.dbf« trägt (Tabelle). Zuerst wird der Inhalt der Datei gelöscht, weil der folgende Append-Befehl die neuen Dateinamen einfach an die Datei anhängen würde.
Sehen Sie sich »dateien.tmp« mit »type« im ASCII-Format an, werden Sie feststellen, daß jede Verzeichniszeile mit Dateinamen, Größe, Datum und Zeit länger als zwölf Zeichen ist. Es interessieren aber nur der Dateiname mit seiner Erweiterung (Extension), und daher werden nur die ersten zwölf Zeichen jeder Zeile eingelesen.
Weil »dir« aber nicht nur die Dateieinträge anzeigt, sondern auch noch den Namen des Laufwerks oder der Festplatte und die Gesamtanzahl der Dateien und den restlichen Speicherplatz auf dem Speichermedium, muß »dateien.dbf« erst einmal gesäubert werden, das heißt, die überflüssigen Zeilen müssen weg. Das sind – wenn Sie zum Beispiel »dir *.prg« eingeben – die ersten vier Datensätze. Drücken Sie bei der Abfrage der Variable »muster« einfach [Enter] für alle Dateien, müssen noch zwei weitere Verzeichniszeilen – nämlich die Datensätze fünf und sechs – entfernt werden, weil der Dir-Befehl bei der Anzeige aller Dateien vor den eigentlichen Dateinamen noch Hinweise auf die Unterverzeichnisverzweigungen

Tabelle. Die Datei »dateien.dbf« für das Ausdrucken von Dateien hat diese Struktur

Feldname	Feldart	Feldlänge
Name	Zeichen	12
Druck	Zeichen	1

».« und »..« gibt, wie Sie bei der Eingabe von »dir« in der MS-DOS-Ebene leicht feststellen können. Sollten noch weitere Unterverzeichnisse gespeichert sein, bleiben diese in »dateien.dbf« vorerst enthalten. Wenn Sie etwas dagegen haben, bauen Sie noch eine weitere Routine ein, die die Einträge auf die Extension »dir« untersucht.
Die folgende Do-While-Schleife fragt den Benutzer nun bei jeder Datei, ob sie gedruckt werden soll oder nicht, und speichert die Antwort »j« oder »n« im Feld »druck«. Nach der Frage, ob der Drucker eingeschaltet ist, sorgt ein »set filter« dafür, daß für den Rest des Programms nur noch die Dateinamen beachtet werden, die wirklich gedruckt werden sollen. Die folgende Schleife, die die Dateien nacheinander, mit jeweils einem Seitenvorschub zwischen den einzelnen Dateien, ausdruckt, muß aus jedem Eintrag in »dateien.dbf« erst einmal einen gültigen Dateinamen herstellen.
Das liegt an der Eigenart, daß beim Dir-Befehl Dateiname und Extension nicht, wie üblich, durch einen Punkt, sondern durch einen Freiraum getrennt sind. Außerdem gibt »dir« die Dateinamen schön formatiert aus, das heißt, Dateinamen und Extensionen stehen hübsch in Spalten untereinander, zum Beispiel

```
ausdruck prg
dateien prg
123 prg
```

Doch mit

```
type 123 prg to print
```

weiß dBase nichts anzufangen. Der Befehl

```
name = rtrim(substr(name,1,8)) + "." + substr(name,10,3)
```

sorgt dafür, daß der genannte Verzeichniseintrag zu dem verwendbaren Dateinamen

```
123.PRG
```

wird. Mit diesen Erläuterungen sollte es für Sie nun einfach sein, in Ihre Anwendungen ähnliche Routinen mit Verzeichnisabfragen einzubauen. (Ulrich Spranger/al)

dBase besorgt die Kalenderwoche

Geschäftstermine werden nicht nur nach Tag und Stunde festgelegt, sondern auch nach Wochen. Eine Kalenderwoche nicht nur nach dem Kalender zu bestimmen, sondern auch in Programmen berechnen zu lassen, ist das Anliegen der Listings 2 und 3. Weil zur Bestimmung einer Kalenderwoche internationale Vereinbarungen bestehen, die hierzulande in der DIN-Norm 1355 festgelegt sind, können Algorithmen verwendet werden, die nicht nur ein einzelnes Jahr betreffen. Unklar ist weithin die Festlegung der ersten Kalenderwoche im Jahr. Ferner muß die Berücksichtigung der Schaltjahre dazugehören. Ansonsten ist nämlich die Zahl der Tage im Monat fixiert. Die Zahl und der Zeitraum der Wochen im Jahr läßt sich damit bestimmen.

Die Festlegung des Wochenbeginns über den Jahreswechsel hinweg geht nicht davon aus, daß der erste Montag im Jahr die erste Kalenderwoche einläutet, sondern der erste Donnerstag. Weil eine Woche nicht am Jahresende getrennt wird, kann es sein, daß die alten Tage des Jahres noch zur ersten Woche des nächsten Jahres zählen – je nachdem, wie viele Tage daran beteiligt sind: Die Woche, in die der Jahreswechsel fällt, gehört zu dem Jahr, in dem sie zum größeren Teil liegt. Da als Wochenanfang Montag gilt, ist also der erste Donnerstag im Jahr der »Kalenderwochen-Teiler« – die betreffende Woche erhält die Nummer eins. Denn von Donnerstag bis Sonntag sind es vier, von Montag bis Mittwoch nur drei Tage.

Ab und zu gibt es auf diese Weise sogar 53 Wochen – dann liegen die ersten Tage des Jahres noch in der letzten Kalenderwoche des Vorjahres. Das Vorjahr hat nämlich immer dann auch eine 53. Woche, wenn im laufenden Jahr der 1.1. ein Freitag oder, nach einem Schaltjahr, ein Samstag ist.

Das Demoprogramm in Listing 2 ist nur für die Verifizierung von Listing 3 gedacht, das den eigentlichen Kalendercode aufarbeitet. Dabei kann das Programm nicht nur die Kalenderwoche zu einem bestimmten Termin zuordnen, sondern auch die einzelnen Kalenderwochen abstecken (Bild). Sie wählen also beispielsweise die Kalenderwoche 7 im Jahre 1990 und erfahren das Datum ihres Beginns (Montag) und das Datum des Sonntags an ihrem Ende. (Friedel Laudon/al)

```
[482]    1: * AUSDRUCK.PRG
[3D6]    2: * (c) Redaktionsbüro Everts & Hagedorn
[4L2]    3: muster = "                "
[3VØ]    4: clear
[P22]    5: set talk off
[K12]    6: set echo off
[FD2]    7: set status off
[Z12]    8: set bell off
[7N8]    9: @ 2,Ø say "                    ---- Auswahl der D
            ateien zum Druck ----- "
[GX9]   1Ø: @ 4,Ø say "Im folgenden wird bei der selekt
            iven Auswahl Ihre "
[C47]   11: @ 5,1Ø say " Wildcard-Liste berücksichtigt:
            "
[KMB]   12: @ 7,Ø say "Bitte geben Sie ein, welche Date
            ien in Frage kommen: " get muster
[VNØ]   13: read
[A34]   14: !dir &muster > dateien.tmp
[G22]   15: use dateien
[BA1]   16: dele all
[UNØ]   17: pack
[AT4]   18: append from dateien.tmp sdf
[2D2]   19: delete record 1
[6D2]   2Ø: delete record 2
[AD2]   21: delete record 3
[ED2]   22: delete record 4
[BZ2]   23: if muster = "                "
[KD2]   24:   delete record 5
[PD2]   25:   delete record 6
[TUØ]   26: endif
[GG1]   27: go bottom
[321]   28: delete
[UNØ]   29: pack
[XYØ]   3Ø: go top
[DX2]   31: do while .not. eof()
[6L6]   32:    @ 8,Ø say name + "  drucken (j/n)? " get
             druck
[VNØ]   33:    read
[VRØ]   34:    skip
[6VØ]   35: enddo
[W81]   36: wahl = " "
```

```
[7P7]   37: @ 1Ø,Ø say "Drucker eingeschaltet (j/n)? "
            get wahl
[VNØ]   38: read
[LX1]   39: if wahl <> "j"
[XA2]   4Ø:    close database
[651]   41:    return
[TUØ]   42: endif
[S35]   43: set filter to upper(druck) = "J"
[XYØ]   44: go top
[DX2]   45: do while .not. eof()
[Ø58]   46:    name = rtrim(substr(name,1,8)) + "." + s
            ubstr(name,1Ø,3)
[W73]   47:    type &name to print
[7VØ]   48:    eject
[VRØ]   49:    skip
[6VØ]   5Ø: enddo
[XA2]   51: close database
[651]   52: return
```

Listing 1. »ausdruck.prg« hilft beim Drucken mehrerer Programme

```
[GS2]    1: PUBLIC xmoncurr
[ØS2]    2: PUBLIC xmonprev
[NS2]    3: PUBLIC xmonnext
[P22]    4: SET TALK OFF
[GB4]    5: SET PROCEDURE TO kw_proc
[D84]    6: ****** Bildschirmaufbau ******
[3VØ]    7: CLEAR
[ØE2]    8: @  Ø, Ø SAY DATE()
[EC6]    9: @  Ø,27 SAY "Rechnen mit Kalenderwochen"
[EV3]   1Ø: @  2,35 SAY "Auswahlmenü"
[S33]   11: @  3, Ø TO  3,79 DOUBLE
[GX1]   12: @ 2Ø, Ø TO 2Ø,79
[MV1]   13: DO WHILE .T.
[WK1]   14:   wahl     = " "
[TG1]   15:   xdiff    = Ø
[ØD3]   16:   xdatum   = CTOD("  .  .  ")
[2H1]   17:   xkwn     = Ø
```

```
[1M2]    18:    xkwc      = SPACE(5)
[5W2]    19:    @  2, Ø  CLEAR TO  2,79
[YV3]    2Ø:    @  2,35  SAY "Auswahlmenü"
[KY2]    21:    @  5, Ø  CLEAR TO 18,79
[LØ3]    22:    @ 21, Ø  CLEAR TO 21,79
[RS3]    23:    @  9,33  SAY "1  Datum in KW"
[PØ4]    24:    @ 11,33  SAY "2  KW in Datum"
[NR3]    25:    @ 15,33  SAY "RET  -->  Ende"
[AZ4]    26:    @ 21, Ø  SAY "Bitte wählen Sie..."
[1T2]    27:    SET CONSOLE OFF
[6D2]    28:    WAIT " " TO wahl
[1H2]    29:    SET CONSOLE ON
[KY2]    3Ø:    @  5, Ø  CLEAR TO 18,79
[LØ3]    31:    @ 21, Ø  CLEAR TO 21,79
[761]    32:    DO CASE
[MP3]    33:    CASE LEN(TRIM(wahl)) = Ø
[ASØ]    34:      EXIT
[P62]    35:    CASE wahl = "1"
[5W2]    36:      @  2, Ø  CLEAR TO  2,79
[UK3]    37:      @  2,35  SAY "Datum in KW"
[2N5]    38:      @ 1Ø,1Ø  SAY "Geben Sie ein Datum ein"
[S83]    39:      @ 1Ø,34  SAY "<TTMMJJ>"
[9X2]    4Ø:      @ 1Ø,44  GET xdatum
[VNØ]    41:      READ
[SN1]    42:      DO montage
[SM1]    43:      DO date2kw
[3P1]    44:      kwc = xkwc
[GS1]    45:      kwn = xkwn
[4G5]    46:      @ 13, Ø  SAY "Die Kalenderwoche zum"
[RX2]    47:      @ 13,22  SAY xdatum
[DB2]    48:      @ 13,31  SAY "ist"
[K82]    49:      @ 13,35  SAY kwc
[1W3]    5Ø:      @ 13,42  SAY "Numerisch : "
[GH3]    51:      @ 13,54  SAY STR(kwn,4,Ø)
[T62]    52:    CASE wahl = "2"
[5W2]    53:      @  2, Ø  CLEAR TO  2,79
[TM3]    54:      @  2,35  SAY "KW in Datum"
[UE5]    55:      @ 1Ø,1Ø  SAY "Geben Sie eine KW ein"
[6V2]    56:      @ 1Ø,32  SAY "<WWJJ>"
[E44]    57:      @ 1Ø,39  GET xkwc PICT "99/99"
[VNØ]    58:      READ
[7K5]    59:      xdatum = CTOD("Ø1.Ø1." + RIGHT(xkwc,2))
[SN1]    6Ø:      DO montage
[CK1]    61:      DO kw2date
[L87]    62:      IF xdatum > xmonnext .OR. xdatum = xmon
           next
[HK8]    63:        @ 13, Ø  SAY "Die angegebene Kalender
           woche ist zu gross"
[ZRØ]    64:      ELSE
[BM5]    65:        @ 13, Ø  SAY "Die KW " + xkwc + " ist
            vom "
[VX2]    66:        @ 13,23  SAY xdatum
[BE2]    67:        @ 13,32  SAY "bis "
[SB3]    68:        @ 13,36  SAY xdatum + 6
[TUØ]    69:      ENDIF
[PA1]    7Ø:    ENDCASE
[PV4]    71:    @ 21, Ø  SAY "Taste drücken..."
[E32]    72:    SET CONS OFF
[9RØ]    73:    WAIT
[822]    74:    SET CONS ON
[6VØ]    75: ENDDO
[651]    76: RETURN
```

Listing 2. »kw__demo.prg« ist eine Demonstrationsoberfläche für die Routinen von »kw__proc.prg«

Name	kw_proc.prg
Sprache	dBase III PLus
Grafikkarte	jede
Besonderheiten	zum Einbau in eigene Routinen. Die Demonstration läuft mit dem Modul »kw_demo.prg«

```
[BZ2]     1: PROCEDURE montage
[YA2]     2: * übergeben :
[H46]     3: * xdatum  = zu berechnendes Datum     (D)
[EN1]     4: * zurück:
[196]     5: * xmonprev= 1. Montag letztes Jahr    (D)
[4C6]     6: * xmoncurr= 1. Montag laufendes Jahr  (D)
[9F6]     7: * xmonnext= 1. Montag kommendes Jahr  (N)
[CB3]     8: xjahr   = YEAR(xdatum)
[PM3]     9: xjahrc = STR(xjahr,4,Ø)
[H73]    1Ø: datØ    = "Ø1.Ø1."+xjahrc
[XR2]    11: datØ    = CTOD(datØ)
[2G2]    12: tagØ    = DOW(datØ)
[761]    13: DO CASE
[TW1]    14: CASE tagØ = 2
[TH2]    15:   xmoncurr = datØ
[VW1]    16: CASE tagØ = 3
[RY2]    17:   xmoncurr = datØ - 1
[XW1]    18: CASE tagØ = 4
[VY2]    19:   xmoncurr = datØ - 2
[ZW1]    2Ø: CASE tagØ = 5
[ZY2]    21:   xmoncurr = datØ - 3
[1X1]    22: CASE tagØ = 6
[VY2]    23:   xmoncurr = datØ + 3
[3X1]    24: CASE tagØ = 7
[RY2]    25:   xmoncurr = datØ + 2
[RW1]    26: CASE tagØ = 1
[LY2]    27:   xmoncurr = datØ + 1
[PA1]    28: ENDCASE
[HT3]    29: xjahrc = STR(xjahr-1,4,Ø)
[H73]    3Ø: datØ    = "Ø1.Ø1."+xjahrc
[XR2]    31: datØ    = CTOD(datØ)
[2G2]    32: tagØ    = DOW(datØ)
[761]    33: DO CASE
[TW1]    34: CASE tagØ = 2
[UG2]    35:   xmonprev = datØ
[VW1]    36: CASE tagØ = 3
[SX2]    37:   xmonprev = datØ - 1
[XW1]    38: CASE tagØ = 4
[WX2]    39:   xmonprev = datØ - 2
[ZW1]    4Ø: CASE tagØ = 5
[ØY2]    41:   xmonprev = datØ - 3
[1X1]    42: CASE tagØ = 6
[WX2]    43:   xmonprev = datØ + 3
[3X1]    44: CASE tagØ = 7
[SX2]    45:   xmonprev = datØ + 2
[RW1]    46: CASE tagØ = 1
[MX2]    47:   xmonprev = datØ + 1
[PA1]    48: ENDCASE
[9T3]    49: xjahrc = STR(xjahr+1,4,Ø)
[H73]    5Ø: datØ    = "Ø1.Ø1."+xjahrc
[XR2]    51: datØ    = CTOD(datØ)
[2G2]    52: tagØ    = DOW(datØ)
[761]    53: DO CASE
[TW1]    54: CASE tagØ = 2
[TH2]    55:   xmonnext = datØ
[VW1]    56: CASE tagØ = 3
[RY2]    57:   xmonnext = datØ - 1
[XW1]    58: CASE tagØ = 4
[VY2]    59:   xmonnext = datØ - 2
[ZW1]    6Ø: CASE tagØ = 5
[ZY2]    61:   xmonnext = datØ - 3
[1X1]    62: CASE tagØ = 6
[VY2]    63:   xmonnext = datØ + 3
[3X1]    64: CASE tagØ = 7
[RY2]    65:   xmonnext = datØ + 2
[RW1]    66: CASE tagØ = 1
[LY2]    67:   xmonnext = datØ + 1
[PA1]    68: ENDCASE
[651]    69: RETURN
[834]    7Ø: ****************************************
[4W2]    71: PROCEDURE date2kw
[YA2]    72: * übergeben :
[256]    73: * xdatum   = umzurechnendes  Datum     (D)
[C66]    74: * xmonprev = 1. Montag letztes Jahr    (D)
[L66]    75: * xmoncurr = 1. Montag laufendes Jahr  (D)
[A96]    76: * xmonnext = 1. Montag kommendes Jahr  (N)
[EN1]    77: * zurück:
[E96]    78: * xkwn  = Kalenderwoche num.  (N) >JJWW<
[5Z5]    79: * xkwc  =  Kalenderwoche chr. (C) >WW/JJ<
[9A4]    8Ø: kwjahr = YEAR(xdatum) - 19ØØ
[761]    81: DO CASE
[6X3]    82: CASE xdatum = xmonnext
[BTØ]    83:   kw = 1
[A23]    84:   kwjahr = kwjahr + 1
[8X3]    85: CASE xdatum > xmonnext
[BTØ]    86:   kw = 1
[A23]    87:   kwjahr = kwjahr + 1
[4X3]    88: CASE xdatum < xmonnext
[TK9]    89:   kw = INT(((xmonnext - xmoncurr) - (xmonne
           xt - xdatum)) / 7) + 1
[SV3]    9Ø: CASE xdatum = xmoncurr
[BTØ]    91:   kw = 1
[PV3]    92: CASE xdatum < xmoncurr
[CM4]    93:   kw = (xmoncurr - xmonprev) / 7
[E23]    94:   kwjahr = kwjahr - 1
[PA1]    95: ENDCASE
[Z91]    96: IF kw < 1Ø
[UR2]    97:   kw = "Ø" + STR(kw,1,Ø)
[ZRØ]    98: ELSE
[XD2]    99:   kw = STR(kw,2,Ø)
[TUØ]   1ØØ: ENDIF
[S84]   1Ø1: xkwc   = STR(kwjahr,2,Ø) + kw
[6S2]   1Ø2: xkwn   = VAL(xkwc)
[HØ5]   1Ø3: xkwc   = kw + "/" + STR(kwjahr,2,Ø)
[651]   1Ø4: RETURN
[W54]   1Ø5: ****************************************
[GW2]   1Ø6: PROCEDURE kw2date
[YA2]   1Ø7: * übergeben :
[X16]   1Ø8: * xkwc = umzurechnende KW    (C) >WW/JJ<
[686]   1Ø9: * xmonprev= 1. Montag letztes Jahr    (D)
[9B6]   11Ø: * xmoncurr= 1, Montag laufendes Jahr  (D)
[GD6]   111: * xmonnext= 1. Montag kommendes Jahr  (N)
[L66]   112: * zurück    ; xdatum  = Datum zur KW  (N)
[KB3]   113: xjahr  = RIGHT(xkwc,2)
[9N3]   114: kw     = VAL(LEFT(xkwc,2))
[DE2]   115: tage   = (kw * 7) - 7
[MW3]   116: xdatum = xmoncurr + tage
[651]   117: RETURN
```

Listing 3. »kw__proc.prg« besteht aus zwei Routinen, die die Kalenderwochen festlegen – nach Termin und Lage im Jahr

Fragen und Antworten zu Quick Basic

Auf dieser Seite finden Sie einige Tips zum Quick-Basic-Compiler von Microsoft. Bestimmt sind auch Antworten zu Problemen dabei, mit denen Sie häufig zu kämpfen haben. Vielleicht möchten Sie beispielsweise Ihre C-Programme zur Zusammenarbeit mit Quick Basic bewegen?

Frage: Kann ich den zum Compiler mitgelieferten Treiber für Hercules-Grafikkarten in meine eigenen Programme einbauen? Woher kann ich den Quelltext des Treibers beziehen, um in eigenen Routinen direkt die Hercules-Karte anzusprechen?

Antwort: Der Hercules-Treiber kann weder in ein Programm eingebaut werden, noch ist er im Quellcode verfügbar. Dieser Treiber zur Unterstützung bei Grafikbefehlen in Quick Basic (Dateiname »msherc.com«, früher »qbherc.com«) ist nicht, wie viele Anwender häufig vermuten, ein Initialisierungsprogramm für die Hercules-Karte. Vielmehr handelt es sich hierbei um ein speicherresidentes Programm (»terminate and stay resident«, auf Deutsch: »beende das Programm ordnungsgemäß, aber lasse es resident im Arbeitsspeicher«).
Theoretisch wäre es möglich, entsprechende Unterstützung der Hercules-Karte für alle Grafikbefehle gleich in die Grafikbibliotheken von Quick Basic einzubinden. Dies würde die Bibliotheken jedoch nur unnötig »aufblähen«, weil bei jedem Aufruf einer BIOS-Grafikfunktion eine Fallunterscheidung für Hercules-Karten erfolgen müßte. Denn im Gegensatz zu den CGA- und EGA-Modi bietet das BIOS eines PC keine direkte Unterstützung des Hercules-Standards.
Der Hercules-Treiber von Quick Basic ist im Prinzip nichts anderes als ein CGA-Emulator. Quick Basic verhält sich mit diesem Emulator so, als ob Sie eine »echte« CGA-Karte in Ihren Rechner eingebaut haben. Auf die Farbmöglichkeiten des CGA-Modus müssen Sie dabei allerdings verzichten, die Farben werden in verschiedene Punktraster übersetzt.

Frage: Wie übergibt man aus einer Basic-Funktion einen String an ein Unterprogramm in C?

Antwort: Die folgende Basic-Routine »bstrf.bas« (»Basic-String-Funktion«) enthält eine Beispiel-Funktion, die einen String an ein C-Programm zurückliefert. Das C-Programm muß die Basic-Funktion aufrufen.

```
declare sub csub cdecl()
call csub
end
function basvarfunc$(dummy%)
basvarfunc$ = "Dies ist ein String"
end function
```

Das nun folgende C-Programm »cstref.c« aktiviert die eben beschriebene Basic-Routine, die die Stringvariable »basvarfunc$« zurückgibt. Es wird ein String-Bezeichner eingeführt, um die Daten von der Basic-Funktion zu übernehmen.

```
#include <stdio.h>
struct stringdesc
 {
  int length;
  /* Stringlänge */
  char *string;
  /* Stringadresse */
 };
extern struct stringdesc * pascal basvarfunc(int *dummy);
struct stringdesc *std;
void csub()
 {
  int i;
  std = basvarfunc(0);
  printf("Länge des Strings: %2d\r\n",std->length);
  for(i = 0; i < std->length;i++)
  printf("%c", std->string[i]);
  printf("\r\n");
 }
```

Um diese beiden Programme (C und Basic) in ein lauffähiges Exe-Programm zu übersetzen, sind folgende Aufrufe der beteiligten Compiler beziehungsweise des Linkers notwendig:

```
bc bstrf.bas;
cl /c /am cstrf.c
link /noe bstrf cstrf;
```

Frage: Woran kann es liegen, daß der Quick-Basic-Editor oder damit geschriebene Programme einen Systemabsturz hervorrufen?

Antwort: Die Ursachen eines Systemabsturzes können generell sehr vielfältig sein. Im folgenden werden die Ursachen, jeweils nach Hardware- und Softwarethematik sortiert, aufgezählt und erläutert.

Systemabstürze hervorgerufen durch Software
1. TSR-Programme
Quick-Basic wurde nicht für die Zusammenarbeit mit TSR-Programmen entwickelt. Die Ursachen liegen vor allem in dem integrierten Debugger begründet, weil dieser auf die volle Funktionsfähigkeit der untersten PC-Systemebene angewiesen ist. Um dem Debugger die Möglichkeit zu geben, Systemfehler frühzeitig zu erkennen, mußten die meisten BIOS-Interrupts umgeleitet werden, was unweigerlich zu Problemen mit TSR-Software führt. Diese benützen die BIOS-Interrupts ebenfalls intensiv.
2. Extended/Expanded Memory bei ATs
Quick Basic bietet keine Zusatzfunktionen für Extended oder Expanded Memory. Die entsprechenden Treiber dafür (EMS-Treiber, Memory-Manager und ähnliche) sollten in jedem Fall entfernt werden. Um sicher zu gehen, daß solche Treiber nicht geladen sind, empfehlen wir Ihnen, alle Device-Befehle in der Steuerdatei »config.sys« Ihres Rechners entsprechend kritisch zu beurteilen.

Systemabstürze hervorgerufen durch Hardware:
1. Falsche Einstellungen des Coprozessors
Um Systemabstürze zu finden, die durch falsch gesetzte Optionen verursacht werden, sorgen Sie mit dem MS-DOS-Kommando

```
set no87=xxxx
```

dafür, daß Quick Basic den Coprozessor vorübergehend ignoriert. Sollte Ihr Quick-Basic-Programm dann fehlerfrei laufen, waren die Optionen beim Übersetzen des betreffenden Programms falsch gesetzt. Compilieren Sie das Programm dann erneut.

2. Grafikkarten
Die Grafikkarte Ihres Computers sollte eine Standardkarte (CGA, EGA, VGA oder Hercules) sein. Günstige Einstellungen Ihres Videosystems erreichen Sie in der Regel durch eine der folgenden drei Optionen des MS-DOS-Kommandos »mode«:

- mode mono (für monochrome Bildschirme)
- mode bw80 (für Farbgrafikkarten an einem Schwarz-Weiß-Bildschirm)
- mode co80 (für Farbsysteme und Plasma-Monitore)

Wenn Quick-Basic Ihr Videosystem trotzdem nicht voll unterstützt, versuchen Sie es mit den videospezifischen Optionen »/b« (»black and white«), »/nohi« (»no high-intensity«), »/g« (Refreshrate setzen) oder »/h« (»high resolution«) beim Start der Entwicklungsumgebung.

(zi/tr)

Alle Listings auf Datenträger erhältlich

Jetzt auch im Abo

DOS EXTRA

Immer wieder aktuell

- 3'88 Der PC - glasklar
- 4'88 MS-DOS intern
- 5'88 Von Basic bis BIOS
- 6'89 Alles über Basic
- 7'89 Power für Programmierer
- 8'89 Assembler von Anfang an

Alle Hefte und Databoxen sind direkt über den Verlag zu beziehen. DOS Extra 3 bis 9

18,– DM (unverbindliche Preisempfehlung)			
Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:			
Inland:		**Ausland:**	
Einzelpreis	18,– DM	Einzelpreis	18,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**22,– DM**	**Endpreis**	**24,– DM**

Bitte benutzen Sie die Bestellkarte.

Databox-Inhalte der DOS Extra Nr. 9

- SCheck 2.01: Checksummer für geprüfte Eingaben
- Definitive Eingabe für Batchdateien
- Minidatenbank per Batchdatei
- Utility für komfortable MS-DOS-Befehlseingabe
- Utility für sicheres Dateilöschen
- Pulldown-Menüs unter GW-Basic programmieren
- Briefbogen selbstgemacht
- Analoguhr für Selfmade-Designer
- Konvertierung vom EGA- in den Hercules-Modus: Schraffur statt Farbe
- Sowie die begleitenden Listings aller anderen Beiträge

DOS TASCHENBÜCHER

Das "goldene" DOS Extra 1 ist wieder erhältlich!

- als Taschenbuch immer griffbereit

Anstelle des ausverkauften, legendären Sonderheftes DOS Extra 1 bringt Ihnen das erste DOS-Taschenbuch alle wichtigen Beiträge, Illustrationen, Tabellen und Listings in neuer ergonomischer Form - platzsparend und immer griffbereit.

DOS-Taschenbuch Band 1

Unter dem Titel "Von Assembler bis MS-DOS" finden Sie in dem neuen DOS-Taschenbuch Band 1 alles Notwendige über die Interna Ihres PCs. Sichern Sie sich ein Exemplar - noch heute!

29,80 DM			
Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:			
Inland:		**Ausland:**	
Einzelpreis	29,80 DM	Einzelpreis	29,80 DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**33,80 DM**	**Endpreis**	**35,80 DM**

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

SCheck 2.02: Checksummer für sichere Listing-Eingabe

Die Programme in der DOS International und der Sonderheftreihe DOS Extra werden nach einem speziellen Verfahren mit Prüfsummen versehen und abgedruckt. Das für die Überprüfung notwendige Prüfsummenprogramm »SCheck« liegt jetzt auch als Version 2.02 für GW-Basic und SBasic vor.

Alle in der DOS International und DOS Extra abgedruckten Programme sind mit Prüfsummen und Zeilennummern versehen. Jede Programmzeile beginnt mit einer dreistelligen Prüfsumme. Sie ist in eckige Klammern gesetzt und wird beim Abtippen nicht mit eingegeben. Nach der Klammer folgt ein Leerzeichen, und darauf der eigentliche Text der Programmzeile (nur bei Basic-Programmen) oder eine Zeilennummer, die Sie ebenfalls nicht mit eingeben, und darauf erst der Programmtext.

Nur diesen letzten Teil der abgedruckten Zeile geben Sie beim Abtippen ein.

Bei allen Programmzeilen wird im Abdruck nach jeweils 55 Zeichen ein Zeilenumbruch vorgenommen. Die umgebrochene Teilzeile erhält dabei keine Prüfsumme. Bei der Eingabe geben Sie umgebrochene Zeilen in einem Stück ein, drücken also nicht die Enter-Taste, wenn ein Zeilenumbruch im Ausdruck steht, sondern nur, wenn eine neue Zeilennummer auftaucht.

Beim Abtippen von Programmen sind Tippfehler kaum zu vermeiden. Mit dem Checksummenprogramm »SCheck« überprüfen Sie die abgetippten Listings auf eventuelle Fehler. Das Programm erlaubt nicht nur die Diagnose von fehlerhaft eingegebenen Zeilen, sondern mittels eines integrierten Editors auch gleich die Korrektur der fehlerhaften Zeilen. SCheck wird unter GW-Basic mit

```
run "scheck"
```

gestartet. Als erstes wählen Sie die gewünschte Prüfsummenversion. In der vorliegenden DOS-Ausgabe brauchen Sie immer »SCheck Version 2.0«. Nur wer Programme aus Ausgaben der DOS International vor Oktober 1988 eingeben will, muß den ebenfalls integrierten Prüfsummenalgorithmus »SCheck Version 1.x« aufrufen. Danach geben Sie den gewünschten Prüfmodus – normalerweise »Standard« – an und abschließend den Programmnamen des Listings, so wie Sie es auf dem Datenträger abgelegt haben.

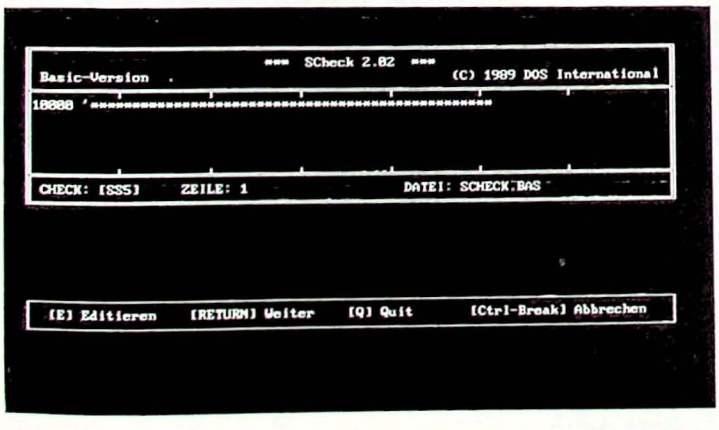
Achtung, Basic-Listings müssen dabei im ASCII-Format vorliegen und dürfen nicht im Token-Format gespeichert sein. Geben Sie dazu im GW-Basic-Interpreter beim Speichern des Programms »name« den Befehl

```
SAVE"name",A
```

ein, um dieses Format zu erzwingen. Zuvor sollten Sie sich aber sicherheitshalber eine Kopie Ihres Listings anfertigen. SCheck arbeitet zwar nicht mit dem Original, sondern mit einer temporären Zwischendatei und verändert die Originaldatei erst ganz zum Schluß, nach dem erfolgreichen Abschluß der Arbeit. Dennoch sollten Sie auch ein noch so geringes Risiko eines Datenverlusts, etwa durch Fehler beim Speichern, ausschalten. Wie Sie im Bild sehen, zeigt SCheck die aktuelle Zeile stets in einem besonderen Bildschirmfenster an. Die Prüfsumme erscheint links unter dem Fenster. Beim Überprüfen Ihres Listings vergleichen Sie die angezeigte Prüfsumme mit der in der DOS-Ausgabe in eckigen Klammern abgedruckten Prüfsumme. Rechts daneben finden Sie die aktuelle Zeilennummer und den Dateinamen. Im unteren Fenster werden Fehlermeldungen und Hinweise auf die weitere Bearbeitung ausgegeben. Im abgebildeten Fall können Sie mit [Enter] die Zeile übernehmen oder mit [E] den Editor aufrufen.

Der Editor dient der Korrektur fehlerhafter Zeilen. Er arbeitet immer im Einfügemodus, so daß fehlerhafte Eingaben extra gelöscht werden müssen. Sie können also nichts versehentlich überschreiben. Mit den Cursor-Tasten steuern Sie den Cursor im Editierfenster. [Home] bringt den Cursor an den Anfang der angezeigten Zeile, [End] entsprechend ans Ende.

Löschen können Sie das Zeichen links vom Cursor mit der großen Löschtaste oberhalb der Enter-Taste, [Del] entfernt das Zeichen unter dem Cursor. Mit [Enter] übernehmen Sie die korrigierte Zeile. Steuerzeichen wie zum Beispiel Tabulatoren dürfen in einer Zeile nicht verwendet werden. Haben Sie so das gesamte Programm oder auch nur »verdächtige« Teile davon durchgearbeitet, so speichert ein letzter Druck auf die Enter-Taste das korrigierte Listing. SCheck wird beendet, und das fehlerfreie Programmlisting steht zu Ihrer Verfügung.



Der Arbeitsbildschirm von SCheck 2.02 in Aktion

Wenn Sie die Prüfsummen mit abtippen, kann SCheck Ihr Listing automatisch überprüfen

SCheck erlaubt aber noch einen zweiten Weg, das Listing korrekt einzutippen. Dabei geben Sie die Prüfsumme gleich bei der Eingabe ein. Die eckigen Klammern lassen Sie aber weg. Hinter der dreistelligen Prüfsumme fügen Sie ein Leerzeichen ein und danach den normalen Programmtext dieser Zeile. Wenn Sie den Text so aufbereitet vorliegen haben, können Sie die automatische Überprüfung mit SCheck nutzen. Rufen Sie dazu die Option »Mit Prüfsumme« auf.

SCheck verwandelt das Listing nun in eine Quelldatei für den jeweiligen Compiler oder Interpreter, indem die drei Prüfsummenstellen und das Leerzeichen vom Rest der Zeile abgetrennt werden. Aus der so entstandenen Programmzeile berechnet SCheck die Prüfsumme und ver-

gleicht sie mit der zuvor abgetrennten Summe, die Sie eingegeben haben. Falls ein Unterschied auftritt, haben Sie Gelegenheit, die fehlerhafte Zeile manuell zu korrigieren.
Nach einem solchen Durchlauf von SCheck steht wieder das fehlerfreie Programm, diesmal allerdings ohne Prüfsumme, auf der Diskette. Auch hier gilt, daß Sie nicht mit dem Original arbeiten sollten, sondern nur mit einer Kopie, um im Zweifelsfall nochmal von vorne beginnen zu können.
SCheck verfügt aber noch über eine weitere Option. Die von Fall zu Fall abgedruckten hexadezimalen Listings können Sie nämlich mit SCheck im Binärmodus direkt in lauffähige Programme umwandeln. Dazu müssen Sie das Hexdump-Listing mit einem ganz normalen Editor (zum Beispiel Turbo Pascal, Wordstar oder Edlin) als Textdatei eingeben. Anschließend überprüfen Sie mit SCheck zunächst die Korrektheit der abgetippten Daten wie bei jedem anderen Listing. Befindet sich das Programm korrekt auf Diskette, so wird die Option »Hexdump« aufgerufen. SCheck wird nun dazu veranlaßt, eine Hexdump-Textdatei in ein ablauffähiges Programm umzuwandeln. SCheck erzeugt in jedem Fall eine Datei mit der Namenserweiterung »bin«. Um daraus eine lauffähige Datei zu machen, müssen Sie sie mit dem Ren-Befehl von MS-DOS umbenennen:

```
ren <name>.bin <name>.com
```

Falls eine Exe-Datei erzeugt werden soll (bitte entnehmen Sie dies der jeweiligen Programmbeschreibung) lautet der entsprechende Befehl:

```
ren <name>.bin <name>.exe
```

Sie sehen, mit SCheck 2.02 ist das korrekte Abtippen von Programmlistings kein bißchen komplizierter geworden – aber durch einen verbesserten Prüfsummen-Algorithmus um einiges sicherer. (ev/ma)

SCheck mit SBasic compilieren

Um die Eingabehilfe SCheck mit dem SBasic-Compiler compilieren zu können, müssen Sie lediglich aus Programmzeile 10070 die Zeichenfolge »key off: « löschen und die Zeile 10720 sowie 11240 vollständig entfernen. Schließlich müssen Sie noch die Programmzeile 11170 in

```
11170 GOTO 10750
```

umändern und die Zeile

```
10735 IF EOF(1) THEN 11160
```

neu in das Programm einfügen.

Name	SCheck.bas
Sprache	GW-Basic und SBasic
Grafikkarte	jede
Besonderheiten	Checksummer für Listings in DOS International und DOS Extra

```
[SS5] 1ØØØØ '****************************************
      *******
[F29] 1Ø01Ø '* SCheck - Prüfsummengenerator für DOS-Lis
      tings *
[ML4] 1Ø02Ø '*
            *
[8Ø7] 1Ø03Ø '*            Konzeption: Volker Everts
            *
[AC7] 1Ø04Ø '*          Basic-Version: Markus Zietlow
            *
[XS5] 1Ø05Ø '****************************************
      *******
[4UØ] 1Ø06Ø '
[FL7] 1Ø07Ø KEY OFF: DEFINT A-O,Q-Z: DEFSNG P: COLOR 7,
      Ø,1
[5G8] 1Ø08Ø T = 1: TEMP = 2: F = 3'Ein-/Ausgabekanäle f
      estlegen
[332] 1Ø09Ø FALSE = Ø
[9Ø2] 1Ø1ØØ TRUE = -1
[EY7] 1Ø11Ø ZEICHEN$ = "Ø123456789ABCDEFGHKLMNPRSTUVWXY
      Z"
[9H2] 1Ø12Ø LAUF = FALSE
[6B2] 1Ø13Ø BIN = FALSE
[851] 1Ø14Ø CLS
[9S1] 1Ø15Ø PRINT
[KØ7] 1Ø16Ø PRINT CHR$(2Ø1); STRING$(78, 2Ø5); CHR$(187
      );
[Z5C] 1Ø17Ø PRINT CHR$(186); STRING$(28, 32); "***  SCh
      eck 2.Ø2  ***"; STRING$(29, 32); CHR$(186);
[ØKE] 1Ø18Ø PRINT CHR$(186); " Basic-Version
                                (C) 1989 DOS Internatio
      nal "; CHR$(186);
[C5C] 1Ø19Ø PRINT CHR$(2Ø4); "═══════════════════════
      ═══════════════════════
      ═══"; CHR$(185);
[DR3] 1Ø2ØØ FOR ZAEHLER = 1 TO 5
[ZY6] 1Ø21Ø   PRINT CHR$(186); STRING$(78, 32); CHR$(18
      6);
[CR2] 1Ø22Ø NEXT ZAEHLER
[84C] 1Ø23Ø PRINT CHR$(2Ø4); "═══════════════════════
      ═══════════════════════
      ═══"; CHR$(185);
[ØØD] 1Ø24Ø PRINT CHR$(186); " CHECK:           ZEILE:
                          DATEI:
          "; CHR$(186);
[PØ7] 1Ø25Ø PRINT CHR$(2ØØ); STRING$(78, 2Ø5); CHR$(188
      );
[2D2] 1Ø26Ø LOCATE 2Ø, 1
[B17] 1Ø27Ø PRINT CHR$(218); STRING$(78, 196); CHR$(191
      );
[ØØ7] 1Ø28Ø PRINT CHR$(179); STRING$(78, 32); CHR$(179)
      ;
[C27] 1Ø29Ø PRINT CHR$(192); STRING$(78, 196); CHR$(217
      );
[4V1] 1Ø3ØØ COLOR 7
[9F1] 1Ø31Ø Y = 7
[9V6] 1Ø32Ø LOCATE 7, 31: PRINT "SCheck  Version 1.x"
[PR6] 1Ø33Ø LOCATE 8, 31: PRINT "SCheck  Version 2.Ø"
[M7H] 1Ø34Ø LOCATE 21, 4: PRINT "Bitte wählen Sie mit d
      en Pfeiltasten aus und bestätigen Sie durch [Ente
      r] ";
[2PD] 1Ø35Ø COLOR 14, 1: FOR Q = 3Ø TO 5Ø: LOCATE Y, Q:
       PRINT CHR$(SCREEN(Y, Q)): NEXT Q: COLOR 7, Ø
[UP6] 1Ø36Ø TASTE$ = INKEY$: IF TASTE$ = "" THEN 1Ø36Ø
[P96] 1Ø37Ø IF TASTE$ <> CHR$(Ø) + CHR$(72) THEN 1Ø4ØØ
[4ND] 1Ø38Ø IF Y <> 7 THEN FOR Q = 3Ø TO 5Ø: LOCATE Y,
      Q: PRINT CHR$(SCREEN(Y, Q)): NEXT Q: Y = Y - 1
[532] 1Ø39Ø GOTO 1Ø35Ø
[1A6] 1Ø4ØØ IF TASTE$ <> CHR$(Ø) + CHR$(8Ø) THEN 1Ø43Ø
[1ND] 1Ø41Ø IF Y <> 8 THEN FOR Q = 3Ø TO 5Ø: LOCATE Y,
      Q: PRINT CHR$(SCREEN(Y, Q)): NEXT Q: Y = Y + 1
[232] 1Ø42Ø GOTO 1Ø35Ø
[F35] 1Ø43Ø IF TASTE$ <> CHR$(13) THEN 1Ø36Ø
[8N2] 1Ø44Ø FOR Q = 7 TO 8
[U16] 1Ø45Ø   LOCATE Q, 3Ø: PRINT STRING$(21, 32)
[KN1] 1Ø46Ø NEXT Q
[8Y2] 1Ø47Ø VERSION = Y - 6
[GV1] 1Ø48Ø COLOR 7
[NF1] 1Ø49Ø Y = 7
[MA5] 1Ø5ØØ LOCATE 7, 36: PRINT "Standard"
[4A6] 1Ø51Ø LOCATE 8, 34: PRINT "Mit Prüfsumme"
[C65] 1Ø52Ø LOCATE 9, 37: PRINT "Hexdump"
[V7H] 1Ø53Ø LOCATE 21, 4: PRINT "Bitte wählen Sie mit d
      en Pfeiltasten aus und bestätigen Sie durch [Ente
      r] ";
[9PD] 1Ø54Ø COLOR 14, 1: FOR Q = 3Ø TO 5Ø: LOCATE Y, Q:
       PRINT CHR$(SCREEN(Y, Q)): NEXT Q: COLOR 7, Ø
[3R6] 1Ø55Ø TASTE$ = INKEY$: IF TASTE$ = "" THEN 1Ø55Ø
[4B6] 1Ø56Ø IF TASTE$ <> CHR$(Ø) + CHR$(72) THEN 1Ø59Ø
[BND] 1Ø57Ø IF Y <> 7 THEN FOR Q = 3Ø TO 5Ø: LOCATE Y,
      Q: PRINT CHR$(SCREEN(Y, Q)): NEXT Q: Y = Y - 1
[E32] 1Ø58Ø GOTO 1Ø54Ø
[GA6] 1Ø59Ø IF TASTE$ <> CHR$(Ø) + CHR$(8Ø) THEN 1Ø62Ø
[CND] 1Ø6ØØ IF Y <> 9 THEN FOR Q = 3Ø TO 5Ø: LOCATE Y,
      Q: PRINT CHR$(SCREEN(Y, Q)): NEXT Q: Y = Y + 1
[B32] 1Ø61Ø GOTO 1Ø54Ø
[P35] 1Ø62Ø IF TASTE$ <> CHR$(13) THEN 1Ø55Ø
[GN2] 1Ø63Ø FOR Q = 7 TO 9
[126] 1Ø64Ø   LOCATE Q, 3Ø: PRINT STRING$(21, 32)
[TN1] 1Ø65Ø NEXT Q
[WG4] 1Ø66Ø IF Y = 8 THEN LAUF = TRUE
[NA4] 1Ø67Ø IF Y = 9 THEN BIN = TRUE
[RU5] 1Ø68Ø LOCATE 12, 54: PRINT STRING$(25, 32)
[8DF] 1Ø69Ø GOSUB 1315Ø: LOCATE 21, 12: PRINT "Bitte ge
      ben Sie den Dateinamen der gewünschten Datei ein
      ";
[9P8] 1Ø7ØØ Y = 12: X = 54: LAENGE = 24: MENGE$ = "alle
      ": GOSUB 1Ø89Ø
[N6D] 1Ø71Ø IF (TASTE$ = CHR$(27)) OR (Q$ = "") THEN LO
      CATE 12, 54: PRINT STRING$(25, 32): GOTO 1Ø68Ø
[LM3] 1Ø72Ø ON ERROR GOTO 1116Ø
```

Listing. »SCheck.bas« für GW-Basic

```
[C74] 10730 OPEN Q$ FOR INPUT AS #T
[732] 10740 GOTO 11230
[3Ø2] 10750 COLOR 12
[2A6] 10760 LOCATE 8, 3Ø: PRINT "D A T E I F E H L E R"
[VV1] 10770 COLOR 7
[5GC] 10780 GOSUB 13150: LOCATE 21, 21: PRINT "Drücken
      Sie bitte eine beliebige Taste ";
[B83] 10790 WHILE INKEY$ = ""
[GX2] 10800    PRINT CHR$(7);
[CE1] 10810 WEND
[6U5] 10820 LOCATE 8, 3Ø: PRINT STRING$(21, 32)
[842] 10830 GOTO 10680
[2VØ] 10840 '
[184] 10850 '***********************************
[UD6] 10860 '* Unterprogramm zur Dateneingabe *
[384] 10870 '***********************************
[6VØ] 10880 '
[GG1] 10890 Q$ = ""
[EF2] 10900 LOCATE Y, X
[FS3] 10910 PRINT Q$; CHR$(219);
[GR6] 10920 TASTE$ = INKEY$: IF TASTE$ = "" THEN 10920
[ZK5] 10930 IF (TASTE$ <> CHR$(8)) THEN 10990
[224] 10940    IF LEN(Q$) = Ø THEN 10920
[X44] 10950    Q$ = LEFT$(Q$, LEN(Q$) - 1)
[6N2] 10960    LOCATE Y, X
[V44] 10970    PRINT Q$; CHR$(219); " ";
[8A2] 10980    GOTO 10920
[T98] 10990 IF TASTE$ = CHR$(13) THEN LOCATE Y, X: PRIN
      T Q$; " ";
[639] 11000 IF (TASTE$ = CHR$(13)) OR (TASTE$ = CHR$(27
      )) THEN RETURN
[C55] 11010 IF (ASC(TASTE$) < 32) THEN 10920
[NX4] 11020 IF MENGE$ <> "zahl" THEN 11050
[ZC8] 11030    IF (ASC(TASTE$) < 48) OR (ASC(TASTE$) > 5
      7) THEN 10920
[892] 11040    GOTO 11070
[PU4] 11050 IF MENGE$ = "alle" THEN 11070
[E86] 11060 IF INSTR(MENGE$, TASTE$) = Ø THEN 10920
[YF6] 11070 IF LEN(Q$) < LAENGE THEN Q$ = Q$ + TASTE$
[T22] 11080 GOTO 10900
[6H8] 11090 Y = 10: X = 65: LAENGE = 5: MENGE$ = "zahl"
      : GOSUB 10890
[BV5] 11100 IF (TASTE$ = CHR$(27)) THEN RETURN
[6UØ] 11110 '
[NK2] 11120 '******************
[TR3] 11130 '* Fehler-Routine *
[RK2] 11140 '******************
[AUØ] 11150 '
[KN1] 11160 CLOSE
[ZG2] 11170 RESUME 10750
[DUØ] 11180 '
[8G2] 11190 '******************
[TH3] 11200 '* Hauptprogramm *
[4G2] 11210 '******************
[BUØ] 11220 '
[KP3] 11230 COLOR 7: GOSUB 13150
[N43] 11240 ON ERROR GOTO Ø
[K62] 11250 DATEI$ = Q$
[DF7] 11260 GROSS$ = DATEI$: GOSUB 13040: DATEI$ = GROS
      S$
[3W3] 11270 I = INSTR(DATEI$, ".")
[6K7] 11280 IF I = Ø THEN DATEI$ = DATEI$ + ".": I = LE
      N(DATEI$)
[U64] 11290 EXT$ = MID$(DATEI$, I, 4)
[8T4] 11300 DATEI$ = LEFT$(DATEI$, I - 1)
[HN5] 11310 LOCATE 12, 54: PRINT DATEI$; EXT$
[HB4] 11320 TEMPNAME$ = DATEI$ + ".$$$"
[5E6] 11330 OPEN TEMPNAME$ FOR OUTPUT AS #TEMP
[CZ7] 11340 IF BIN THEN OPEN DATEI$ + ".BIN" FOR OUTPUT
       AS #F
[CP7] 11350 IF VERSION = 2 THEN NUMMER = 1 ELSE NUMMER
      = Ø
[KYC] 11360 IF BIN THEN GOSUB 13150: LOCATE 21, 25: PRI
      NT ">>> Binärdatei wird erzeugt <<<"
[TE3] 11370 WHILE NOT EOF(T)
[X24] 11380    LINE INPUT #T, ZEILE$
[LDB] 11385    WHILE RIGHT$(ZEILE$,1)=" " : ZEILE$=LEFT$
      (ZEILE$,LEN(ZEILE$)-1) : WEND
[333] 11390    WEITER = FALSE
[WT3] 11400    IF WEITER THEN 11500
[WN6] 11410    GOSUB 11820 'Gelesene Zeile anzeigen
[TS4] 11420    IF BIN THEN WEITER = TRUE
[ZT3] 11430    IF WEITER THEN 11500
[AM6] 11440    CH$ = INKEY$: IF CH$ = "" THEN GOTO 11440
[3G3] 11450    ERLAUBT = FALSE
[ZH8] 11460    IF CH$ = CHR$(13) THEN ERLAUBT = TRUE: WE
      ITER = TRUE
[N1C] 11470    IF (CH$ = "e") OR (CH$ = "E") THEN ERLAUB
      T = TRUE: GOSUB 12060'Zeile bearbeiten
[8ME] 11480    IF (CH$ = "q") OR (CH$ = "Q") THEN IF NOT
       LAUF THEN GOSUB 12420: IF SCHLUSS THEN GOTO 1163
      Ø'Quit
[F57] 11490    IF ERLAUBT THEN GOTO 11400 ELSE GOTO 1144
      Ø
[8Y3] 11500    PRINT #TEMP, ZEILE$
[DU3] 11510    IF NOT BIN THEN 11610
[D99] 11520    WHILE LEFT$(ZEILE$, 1) = " ": ZEILE$ = MI
      D$(ZEILE$, 2): WEND
[364] 11530    IF ZEILE$ = "" THEN 11610
[K75] 11540    B = VAL("&H" + LEFT$(ZEILE$, 2))
[TSH] 11550    IF (B < Ø) OR (B > 255) THEN GOSUB 13150:
       LOCATE 21, 22: PRINT "+++++ ABBRUCH WEGEN DATENF
      EHLER +++++": LOCATE 25, 1: END
[HN1] 11560    BY = B
[TK3] 11570    PRINT #F, CHR$(BY);
[YA4] 11580    ZEILE$ = MID$(ZEILE$, 3)
[M99] 11590    WHILE LEFT$(ZEILE$, 1) = " ": ZEILE$ = MI
      D$(ZEILE$, 2): WEND
[T92] 11600    GOTO 11530
[DY3] 11610    NUMMER = NUMMER + 1
[8E1] 11620 WEND
[S7B] 11630 WHILE NOT EOF(T): LINE INPUT #T, ZEILE$: PR
      INT #TEMP, ZEILE$: WEND
[4P1] 11640 CLOSE
[8U5] 11650 OPEN DATEI$ + EXT$ FOR INPUT AS #1
[XØ6] 11660 OPEN DATEI$ + ".BAK" FOR OUTPUT AS #2
[XGB] 11670 WHILE NOT EOF(1): LINE INPUT #1, ZEILE$: PR
      INT #2, ZEILE$: WEND: CLOSE
[6D5] 11680 OPEN TEMPNAME$ FOR INPUT AS #1
[576] 11690 OPEN DATEI$ + EXT$ FOR OUTPUT AS #2
[UGB] 11700 WHILE NOT EOF(1): LINE INPUT #1, ZEILE$: PR
      INT #2, ZEILE$: WEND: CLOSE
[713] 11710 KILL TEMPNAME$
[GTN] 11720 GOSUB 12560: LOCATE 12, 9: PRINT STRING$(11
      , 32): LOCATE 12, 25: PRINT STRING$(21, 32): LOCA
      TE 12, 53: PRINT STRING$(27, 32)'Statuszeile lösc
      hen
[494] 11730 IF BIN THEN EXT$ = ".BIN"
[N4H] 11740 GOSUB 13150: LOCATE 21, (80 - (31 + LEN(DAT
      EI$ + EXT$))) / 2: PRINT "Die Datei '"; DATEI$ +
      EXT$; "' wurde gespeichert."
[CE2] 11750 LOCATE 25, 1
[H51] 11760 END
[4VØ] 11770 '
[TH3] 11780 '****************************
[B65] 11790 '* Gelesene Zeile anzeigen *
[NH3] 11800 '****************************
[2VØ] 11810 '
[BBB] 11820 IF LAUF THEN CHECK$ = "[" + LEFT$(ZEILE$, 3
      ) + "]": ZEILE$ = MID$(ZEILE$, 5)
[MDN] 11830 FOR Q = 1 TO LEN(CHECK$): IF ASC(MID$(CHECK
      $, Q, 1)) > 96 AND ASC(MID$(CHECK$, Q, 1)) < 123
      THEN MID$(CHECK$, Q, 1) = CHR$(ASC(MID$(CHECK$, Q
      , 1)) - 32)
[3P1] 11840 NEXT Q
[B57] 11850 LINKS = 2: OBEN = 6: RECHTS = 79: UNTEN = 1
      Ø
[CX4] 11860 LOCATE 12, 26: PRINT NUMMER
[Z45] 11870 GOSUB 12560'Fenster löschen
[WX8] 11880 GOSUB 12660'Zeile innerhalb des Fensters an
      zeigen
[366] 11890 GOSUB 12800'Check-Summe errechnen
[ØM2] 11900 LOCATE 12, 10
[HB2] 11910 PRINT PS$
[HM7] 11920 IF BIN THEN GOTO 12000'Binärdatei erzeugen
[N8C] 11930 IF LAUF THEN GOTO 11970'Übereinstimmung der
       Prüfsummen feststellen
[X74] 11940 GOSUB 13150: LOCATE 21, 4
[LKD] 11950 PRINT "[E] Editieren    [RETURN] Weiter
      [Q] Quit        [Ctrl-Break] Abbrechen"
[152] 11960 GOTO 11990
[ATP] 11970 GOSUB 13150: IF PS$ = CHECK$ THEN LOCATE 21
      , (80 - (22 + LEN(STR$(NUMMER)))) / 2: PRINT "Prü
      fsumme OK für Zeile"; NUMMER: WEITER = TRUE: GOTO
       12000
[1NH] 11980 LOCATE 21, 4: PRINT CHR$(7); "Prüfsummenfeh
      ler!  [E] Editieren  [RETURN] Weiter  [Ctrl-Break
      ] Abbrechen"
[XM2] 11990 LOCATE 12, 15
[KV1] 12000 RETURN
[5UØ] 12010 '
[MK2] 12020 '******************
[4H3] 12030 '* Zeile eingeben *
[PK2] 12040 '******************
[9UØ] 12050 '
[HRD] 12060 GOSUB 13150: LOCATE 21, 17: PRINT "Ändern S
      ie die Programmzeile        [Enter] Ende"
[245] 12070 GOSUB 12560'Fenster löschen
[7B5] 12080 GOSUB 12660'Zeile darstellen
[FL3] 12090 LOCATE OBEN, LINKS
[7G3] 12100 X = 1: PRINT "";
[8E2] 12110 CH$ = INKEY$
[FE6] 12120 IF LEFT$(CH$, 1) <> CHR$(Ø) THEN GOTO 12220
[4N3] 12130    CH$ = RIGHT$(CH$, 1)
[4NE] 12140    IF CH$ = "M" THEN IF X > LEN(ZEILE$) THEN
       PRINT CHR$(7); : GOTO 12110 ELSE GOSUB 12340: X
      = X + 1
[KZC] 12150    IF CH$ = "K" THEN IF X = 1 THEN PRINT CHR
      $(7); : GOTO 12110 ELSE GOSUB 12340: X = X - 1
[YZC] 12160    IF CH$ = "H" THEN IF X < 79 THEN PRINT CH
      R$(7); : GOTO 12110 ELSE GOSUB 12340: X = X - 78
[YCF] 12170    IF CH$ = "P" THEN IF X + 77 > LEN(ZEILE$)
       THEN PRINT CHR$(7); : GOTO 12110 ELSE GOSUB 1234
      0: X = X + 78
[TV7] 12180    IF CH$ = "G" THEN IF X <> 1 THEN GOSUB 12
      340: X = 1
```

Listing. »SCheck.bas« für GW-Basic

```
[41C] 12190    IF CH$ = "O" THEN IF X <> LEN(ZEILE$) + 1
       THEN GOSUB 12340: X = LEN(ZEILE$) + 1
[8PM] 12200    IF CH$ = "S" THEN IF X <= LEN(ZEILE$) THE
      N ZEILE$ = LEFT$(ZEILE$, X - 1) + MID$(ZEILE$, X
      + 1): GOSUB 12660 ELSE PRINT CHR$(7); : GOTO 1211
      0
[G92] 12210    GOTO 12240
[XXN] 12220 IF CH$ = CHR$(8) THEN IF X > 1 THEN ZEILE$
      = LEFT$(ZEILE$, X - 2) + MID$(ZEILE$, X): GOSUB 1
      2660: GOSUB 12340: X = X - 1 ELSE PRINT CHR$(7);
      : GOTO 12110
[LZN] 12230 IF CH$ >= " " THEN IF LEN(ZEILE$) = 255 THE
      N PRINT CHR$(7); : GOTO 12110 ELSE ZEILE$ = LEFT$
      (ZEILE$, X - 1) + CH$ + MID$(ZEILE$, X): GOSUB 12
      660: X = X + 1
[7YB] 12240 LOCATE OBEN + INT((X - 1) / 78), LINKS - 1
      + X - INT((X - 1) / 78) * 78: PRINT "■":
[LK5] 12250 IF CH$ <> CHR$(13) THEN GOTO 12110
[648] 12260 IF LAUF THEN ZEILE$ = MID$(CHECK$, 2, 3) +
      " " + ZEILE$
[XL6] 12270 GOSUB 12340: GOSUB 12660'Zeile anzeigen
[2W1] 12280 RETURN
[NU0] 12290 '
[CP3] 12300 '****************************
[2H5] 12310 '* Zeichen wiederherstellen *
[EP3] 12320 '****************************
[LU0] 12330 '
[UM9] 12340 LOCATE OBEN + INT((X - 1) / 78), LINKS - 1
      + X - INT((X - 1) / 78) * 78
[YXA] 12350 IF MID$(ZEILE$, X, 1) = "" THEN PRINT " " E
      LSE PRINT MID$(ZEILE$, X, 1)
[4W1] 12360 RETURN
[RU0] 12370 '
[0A2] 12380 '***************
[3X2] 12390 '* Quit (Ende) *
[W92] 12400 '***************
[NU0] 12410 '
[U63] 12420 SCHLUSS = FALSE
[KYB] 12430 SICHER$ = "": FOR Q = 2 TO 79: SICHER$ = SI
      CHER$ + CHR$(SCREEN(21, Q)): NEXT Q
[YDD] 12440 GOSUB 13150: LOCATE 21, 16: PRINT "Bitte dr
      ücken Sie [J], um die Arbeit zu beenden "
[GA6] 12450 CH$ = INKEY$: IF CH$ = "" THEN GOTO 12450
[R3B] 12460 IF (CH$ = "j") OR (CH$ = "J") THEN SCHLUSS
      = TRUE: PRINT #TEMP, ZEILE$
[BA2] 12470 GOSUB 13150
[P78] 12480 IF NOT SCHLUSS THEN LOCATE 21, 2: PRINT SIC
      HER$
[BW1] 12490 RETURN
[651] 12500 END
[TU0] 12510 '
[6R2] 12520 '********************
[U24] 12530 '* Fenster löschen *
[8R2] 12540 '********************
[XU0] 12550 '
[044] 12560 FOR Q = OBEN TO UNTEN
[Z53] 12570 LOCATE Q, LINKS
[8B6] 12580    PRINT STRING$(RECHTS - LINKS + 1, 32)
[ZN1] 12590 NEXT Q
[AW1] 12600 RETURN
[XU0] 12610 '
[265] 12620 '*****************************************
      *
[D98] 12630 '* Inhalt von »zeile$« im Fenster anzeigen
      *
[465] 12640 '*****************************************
      *
[1V0] 12650 '
```

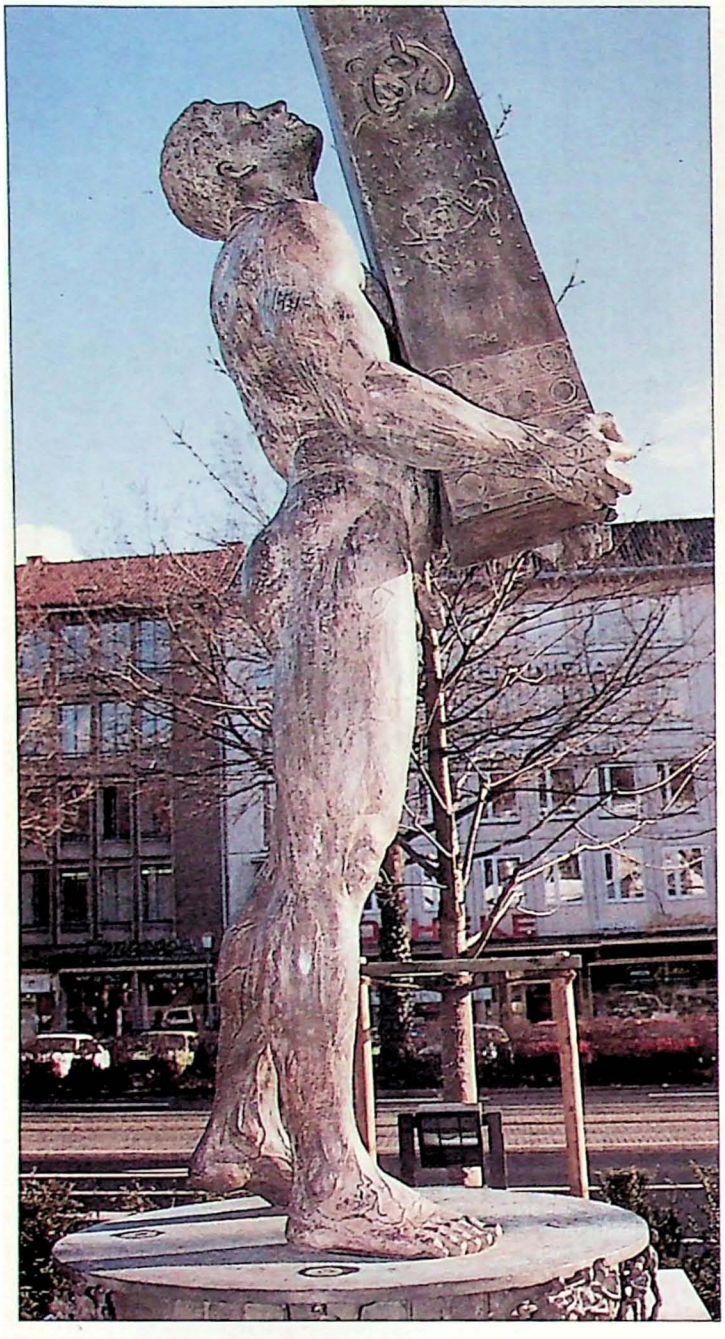
```
[SR2] 12660 F.X = 0: F.Y = 0
[CC5] 12670 FOR ZAEHLER = 1 TO LEN(ZEILE$)
[F85] 12680    LOCATE OBEN + F.Y, LINKS + F.X
[UH5] 12690    PRINT MID$(ZEILE$, ZAEHLER, 1);
[SK2] 12700    F.X = F.X + 1
[597] 12710    IF F.X >= RECHTS - 1 THEN F.X = 0: F.Y =
      F.Y + 1
[6S2] 12720 NEXT ZAEHLER
[RX6] 12730 LOCATE OBEN + F.Y, LINKS + F.X: PRINT " ";
[KW1] 12740 RETURN
[5V0] 12750 '
[AH4] 12760 '***********************************
[ET6] 12770 '* Zeile auf Check-Summe überprüfen *
[CH4] 12780 '***********************************
[9V0] 12790 '
[F42] 12800 P = 0: A = 1
[8SB] 12810 WHILE (MID$(ZEILE$, A, 1) = " ") OR (MID$(Z
      EILE$, A, 1) = CHR$(9)): A = A + 1: WEND
[374] 12820 FOR I = A TO LEN(ZEILE$)
[4CX] 12830    IF MID$(ZEILE$, I, 1) >= " " THEN GROSS$
      = MID$(ZEILE$, I, 1): IF VERSION = 1 THEN P = ((I
       - A + 1) AND 7) * ASC(GROSS$) + P: GOTO 12920 EL
      SE GOSUB 13040: P = (((I - A + 1) AND 3) + 1) * A
      SC(GROSS$) + P: GOTO 12920
[S22] 12840    COLOR 12
[5N6] 12850    LOCATE 8, 30: PRINT "D A T E I F E H L E
      R"
[022] 12860    COLOR 7
[X7C] 12870    LOCATE 21, 11: PRINT "          Drücken S
      ie bitte eine beliebige Taste          ";
[CC3] 12880    WHILE INKEY$ = ""
[D93] 12890       PRINT CHR$(7);
[5L1] 12900    WEND
[R16] 12910    LOCATE 8, 30: PRINT STRING$(21, 32)
[0P1] 12920 NEXT I
[KF2] 12930 PS$ = "[    ]"
[AM2] 12940 FOR I = 2 TO 4
[K25] 12945    IF P > 32767 THEN P = P - 32768
[5T7] 12950    MID$(PS$, I, 1) = MID$(ZEICHEN$, (P AND 3
      1) + 1, 1)
[HZ2] 12960    P = FIX(P / 32)
[5P1] 12970 NEXT I
[YW1] 12980 RETURN
[HV0] 12990 '
[ZX3] 13000 '*******************************
[K06] 13010 '* In Großbuchstaben umwandeln *
[934] 13020 '*******************************
[AU0] 13030 '
[9ZN] 13040 FOR Q = 1 TO LEN(GROSS$): IF ASC(MID$(GROSS
      $, Q, 1)) > 96 AND ASC(MID$(GROSS$, Q, 1)) < 123
      THEN MID$(GROSS$, Q, 1) = CHR$(ASC(MID$(GROSS$, Q
      , 1)) - 32)
[8T8] 13050 IF MID$(GROSS$, Q, 1) = "ä" THEN MID$(GROSS
      $, Q, 1) = "Ä"
[NU8] 13060 IF MID$(GROSS$, Q, 1) = "ö" THEN MID$(GROSS
      $, Q, 1) = "Ö"
[UU8] 13070 IF MID$(GROSS$, Q, 1) = "ü" THEN MID$(GROSS
      $, Q, 1) = "Ü"
[DN1] 13080 NEXT Q
[YV1] 13090 RETURN
[BU0] 13100 '
[YZ2] 13110 '**********************
[9N4] 13120 '* Meldezeile löschen *
[003] 13130 '**********************
[FU0] 13140 '
[ZT5] 13150 LOCATE 21, 2: PRINT STRING$(78, 32)
[ZV1] 13160 RETURN
```

Listing. »SCheck.bas« für GW-Basic

Utilities gesucht

Programme, die den Bildschirminhalt auf den Kopf stellen, suchen unsere Leser sicherlich nicht. Aber einen wissenschaftlichen Taschenrechner, ein Festplatten-Backup-Programm, einen Extended-Memory-Treiber, Druckerspooler oder Cache-Speicher. Zögern Sie deshalb nicht, schicken Sie uns unverbindlich die aktuelle Version Ihres Programms und einen Ausdruck der Dokumentation.
Falls Sie sich vorweg informieren möchten: Zwischen 15.00 und 17.00 Uhr stehen wir Ihnen an allen Werktagen unter der Rufnummer (089) 4 61 00 10 zur Verfügung.

Schicken Sie Ihre Unterlagen an folgende Adresse:
Redaktionsbüro Everts & Hagedorn
»DOS Software«
Hans-Stießberger-Str. 3
8013 Haar/München



Hercules-Grafik in Basic programmiert

Viele Basic-Programmierer besitzen eine Hercules-Grafikkarte, können diese aber im Grafikmodus nicht wie gewünscht programmieren, weil die verschiedenen Basic-Dialekte die Hercules-Grafik nur unzureichend oder überhaupt nicht unterstützen. Damit ist jetzt Schluß! Im folgenden stellen wir Ihnen eine Hercules-Toolbox vor, die für Quick Basic, SBasic und Turbo Basic neue, leistungsfähige Hercules-Grafikroutinen bereitstellt.

Trotz der steigenden Popularität der EGA- und VGA-Karten benutzt immer noch ein Großteil der PC-Besitzer eine Hercules-Grafikkarte. Doch die Grafikfähigkeiten dieser Karte lassen sich kaum oder gar nicht nutzen, wenn man in Basic programmiert, weil die verschiedenen Basic-Dialekte die Hercules-Grafik recht stiefmütterlich behandeln. Aus diesem Grund stellt Ihnen die Hercules-Toolbox »hercules.bas« eine Reihe von leistungsfähigen Grafikroutinen zur Programmierung der Hercules-Karte zur Verfügung, die übrigens auch unter GW-Basic läuft – allerdings entsprechend langsam!

Gleichzeitig enthält das Programm (Listing) auch einige Demonstrationsroutinen, die die Fähigkeiten der Toolbox eindrucksvoll unter Beweis stellen. Interessanterweise ist die Programmausführung dieser Routinen bei der mit SBasic compilierten Version um ein Vielfaches schneller als bei der Quick- oder Turbo-Basic-Version, obwohl der Quelltext nicht speziell für SBasic entwickelt wurde.

Wenn Sie die Hercules-Toolbox starten, wird zunächst auf dem Bildschirm eine Analoguhr aufgebaut, die die aktuelle Zeit anzeigt. In der mit SBasic compilierten Version rückt der Sekundenzeiger der Uhr mit jeder Sekunde um eine Position weiter, so wie man es von einem Sekundenzeiger erwarten kann.

Wird das Grafikprogramm dagegen mit Turbo Basic compiliert und gestartet, so vergehen hin und wieder zwei Sekunden, bevor der Sekundenzeiger weiterrückt. Wird schließlich Quick Basic in der neuesten Version 4.5 verwendet, so vergehen regelmäßig zwei, manchmal sogar drei Sekunden, bevor sich der Sekundenzeiger bewegt (die Geschwindigkeitsangaben beziehen sich auf einen 16-MHz-AT).

Die Hercules-Toolbox ist sehr modular aufgebaut und besteht in ihrer groben Struktur aus vier Programmteilen. Der erste Teil von Programmzeile 1 bis 70 besteht aus den Demonstrationsroutinen, aus denen verschiedene Aktionen wie »Bildschirm löschen«, »Kreis zeichnen« oder »Uhr anzeigen« aufgerufen werden. Je nach Eingabe einer Zahl wird das entsprechende Unterprogramm ausgeführt.

Bezeichnung	Programmzeile	Bedeutung
HercZeige0	71	schaltet Monitor auf Hercules-Grafikseite 0, also auf den Adreßbereich B0000 bis B7FFFhex; dabei werden gesetzt: Herc.Seite.Video% = 0Ahex Herc.Seg.Video% = B000hex Herc.Video.Flag = 0 (Herc.Seg.Video% wird zur Hardware-Steuerung und Herc.Video.Flag% zur Programmsteuerung gesetzt)
Herczeige1	82	schaltet Monitor auf Hercules-Grafikseite 1, also auf den Adreßbereich B8000 bis BFFFFhex; dabei werden gesetzt: Herc.Seite.Video% = 8Ahex Herc.Seg.Video% = B800hex Herc.Video.Flag = 1
HercSchreiben	93	legt Grafikseite fest, in die Daten geschrieben werden Herc.Seg.Schreiben% = B000hex: Seite 0 Herc.Seg.Schreiben% = B800hex: Seite 1
HercLoesch	105	schaltet Hercules-Grafik ein, setzt alle Bits auf Null und schaltet Hercules-Grafik wieder aus
HercHel	113	schaltet Hercules-Grafik ein, setzt alle Bits auf Eins und schaltet Hercules-Grafik wieder aus
HercByte-Setzen	122	fragt die Koordinaten ab und setzt das angegebene Byte
HercByteLoeschen	135	fragt die Koordinaten ab und löscht das angegebene Byte
HercPktSet	148	fragt die Koordinaten ab und setzt den angegebenen Punkt
HercPkt Loesch	161	fragt die Koordinaten ab und löscht den angegebenen Punkt
HercKreis	174	fragt die Koordinaten ab und zeichnet/löscht den gewünschten Kreis
HercLinie	196	fragt die Koordinaten ab und zeichnet/löscht die gewünschte Linie
HercRechteck	218	fragt die Koordinaten ab und zeichnet/löscht das gewünschte Rechteck
HercDreieck	249	fragt die Koordinaten ab und zeichnet/löscht das gewünschte Dreieck
HercBogen	278	fragt die Koordinaten ab und zeichnet/löscht den gewünschten Kreisbogen
HercWinkel	306	fragt die Koordinaten ab und zeichnet/löscht den gewünschten Winkel
HercSeiten	335	verknüpft Grafikseite 0 mit Grafikseite 1 über die And-, Or- oder Xor-Funktion
HercToggel	348	schaltet abwechselnd Grafikseite 0 und Grafikseite 1 auf den Monitor
HercText	361	fragt Text ab und gibt ihn aus
HercUhr	382	zeigt eine Uhr auf dem Monitor an

Tabelle 1. Die Routinen der Hercules-Toolbox zur Anwendung der Grafikunterprogramme

Der zweite Programmteil von Zeile 71 bis 509 enthält die Unterprogramme, die im Demonstrationsteil aufgerufen werden. Diese Unterprogramme erfragen vom Anwender die für den Ablauf der Grafikunterprogramme erforderlichen Werte für die verschiedenen Variablen und starten anschließend diese Unterprogramme.

Die Hercules-Toolbox ist modular aufgebaut und besteht insgesamt aus vier Programmteilen

Der dritte Programmteil von Zeile 510 bis 1089 besteht aus den elementaren Grafikroutinen, die direkt auf die Hardware der Hercules-Grafikkarte zugreifen und die notwendigen Initialisierungen, Bereichsüberprüfungen und Umrechnungen vornehmen. Der vierte und letzte Programmteil von Zeile 1090 bis 1423 enthält schließlich im wesentlichen die Programmroutinen zur Erzeugung einfacher geometrischer Figuren.

VirusDoktor

Eine Programmsammlung, die Sie gegen alle bekannten PC-Computer-Viren wappnet

Auch schon betroffen?

- Schäden durch infizierte, unbrauchbare Programmdateien?
- Datenverlust durch formatierte und zerstörte Speichermedien?
- Zeitverlust und Ärger durch Verstellen der Systemkonfiguration?

TEST SIEGER
PC Plus 9/89

Name: Virusdoktor
Preis: 98 Mark
Anbieter: DMV-Verlag
Kurzinfo: Das Programm checkt zuverlässig und äußerst schnell. Die durchdachte Logik, die CMOS-RAM-Backup-Möglichkeit und der niedrige Preis sprechen ebenfalls für das Programm.
Schutzeffekt: hoch
Killmechanismus: nein
Handhabung: sehr gut

Prädikat
●●●●●●●●●○
1 PLUS-Punkt = ungenügend,
10 PLUS-Punkte = hervorragend

VirCheck -

Ein Kontrollprogramm der Superklasse!

- Kontrolle Ihrer Programme auf Längenänderung, Attributierung etc.
- Spezieller Check für virusbedingte Abänderung des Programmcodes
- Überwachung von Umbenennungen und Verschiebungen von Dateien
- Beliebige Auswahl der kontrollierten Programmgruppen
- Hohe Effektivität und Geschwindigkeit durch variables SETUP
- Komfortables, menügesteuertes, grafisches Installationsprogramm

VirSperr -

dreimal Schutz vor Virusinfektion

- Drei Programme zum Absichern Ihrer Dateien vor Veränderungen
- Ausführlicher Report über versuchte Zugriffe auf Ihre Dateien
- Virussicheres Fixieren von gesetzten Read-Only-Attributen
- Ein- und Ausschalten des Schreibschutzes per Tastendruck

Und als Zugabe

CMOS-BKP - Konfigurationssicherung für AT- und AT- kompatible Rechner

VirusDoktor

Fünf wertvolle Programme zum Preis von einem plus ausführliche Programmanleitung inklusive Grundwissen über Computerviren

Für alle MS-DOS-Computer

99,– DM (unverbindliche Preisempfehlung)
Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	99,– DM	Einzelpreis	99,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**103,– DM**	**Endpreis**	**105,– DM**

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Bezeichnung	Programmzeile	Bedeutung
HercIni	525	keine Parameter, setzt alle Initialisierungswerte und lädt das Alphabet
HercEntwurf	688	dient zum Entwurf von benutzerdefinierten Buchstaben und Zeichen
Warten	708	fährt erst nach Tastendruck fort
Delayx	717	setzt Herc.Delay und erzeugt Verzögerungen für Gosub Delayx
Toggel	724	schaltet zwischen Grafikseite 0 und 1 hin und her
HercEin	743	keine Parameter, schaltet Hercules-Karte ein
HercAus	763	keine Parameter, schaltet Hercules-Karte aus
HercLoeschen	780	keine Parameter, setzt alle Bits der gewählten Hercules-Grafikseite auf Null und löscht dadurch den Bildschirm
HercHell	798	keine Parameter, setzt alle Bits der gewählten Hercules-Grafikseite auf 1 und erzeugt dadurch ein helles Bild
HercBildMode	827	verknüpft die beiden Hercules-Grafikseiten wenn Herc.Seiten.Mode% = 0, dann Seite0 And Seite1 wenn Herc.Seiten.Mode% = 1, dann Seite0 Or Seite1 wenn Herc.Seiten.Mode% = 2, dann Seite0 Xor Seite1
LoescheByte	910	X.B% ist das horizontale Byte (0 bis 79). Y.B% ist das vertikale Byte (0 bis 347).
SetzeByte	936	X.B% ist das horizontale Byte (0 bis 79). Y.B% ist das vertikale Byte (0 bis 347)
ModePunkt	955	Modepunkt% = 1 setzt Punkt Modepunkt% = 2 löscht Punkt Modepunkt% = 2 verknüpft Punkt über die Xor-Funktion mit Bildhintergrund Zuvor wird immer der zulässige Bereich von 0 bis 719 für x und 0 bis 347 für y überprüft. Um einen Punkt zu setzen, müssen X.P% und Y.P% dessen Koordinate enthalten.
Check	966	Diese Routine wird von Modepunkt aufgerufen. Hier können auch Koordinatentransformationen durchgeführt werden (siehe Listing). Diese Routine setzt die beiden Variablen XP.P% und YP.P%, die die Punktkoordinaten enthalten.
Punkt	995	ruft Check und anschließend PunktSet auf
PunktSet	1004	setzt einen Punkt auf dem Bildschirm
XPunkt	1029	ruft Check und anschließend XPunktSet auf
XPunktSet	1038	verknüpft Punkt über die Xor-Funktion mit Bildhintergrund
LPunkt	1056	ruft Check und anschließend PRset auf
PRset	1067	löscht einen Punkt auf dem Bildschirm

Tabelle 2. Routinen der untersten Ebene, die direkt auf die Hardware zugreifen oder vorher notwendige Überprüfungen vornehmen, damit keine Bereichsüberschreitungen auftreten

Die Routinen des zweiten Programmteils sind in Tabelle 1 aufgeführt. Die erste Spalte der Tabelle gibt jeweils den Namen des Unterprogramms, die zweite die Zeilennummer im Quelltext und die dritte Spalte die Bedeutung dieser Routine und die verwendeten Variablen an. In gleicher Weise sind die Routinen des dritten Programmteils in Tabelle 2 und die Routinen des vierten Programmteils in Tabelle 3 aufgeführt.

Die Programmierung der Hercules-Grafikkarte wurde in der DOS International bereits mehrfach beschrieben, so daß die Erklärungen zu den Routinen der Hercules-Toolbox im folgenden recht knapp gehalten sind. Überdies ist das Programm ausführlich dokumentiert, was ebenfalls zum Verständnis der Programmfunktionen beiträgt.

Soll eine Hercules-Grafik auf dem Monitor erscheinen, so muß die Karte zunächst in den Grafikmodus geschaltet werden. Dazu ist folgende Datensequenz an die Karte zu senden:

- In das Konfigurationsregister des Video-Controllers 6845, das über die Portadresse 3BFhex angesprochen

Bezeichnung	Programmzeile	Bedeutung
Kreis	1090	X.K%, Y.K%: Koordinate des Kreismittelpunktes X.R: Kreisradius
Bogen	1117	X.K%, Y.K%: Koordinate des Mittelpunktes X.R: Kreisradius Alpha, Beta: Anfang und Ende des Kreisbogens (Angabe in Grad, Wertebereich von 0 bis 360)
Linie	1143	X.L1%, Y.L1%: Startkoordinate der Linie X.L2%, Y.L2%: Endkoordinate der Linie
HercSpeichern	1235	speichert oder lädt gewählte Hercules-Grafikseite 0 oder 1
HercSatz	1258	Herc.Satz$: auszugebende Zeichenkette Herc.XSatz.Pos%, Herc.YSatz.Pos%: Startkoordinate für auszugebenden Text Herc.Satz.Mode% = 0: Satz horizontal ausgeben Herc.Satz.Mode% = 1: Satz vertikal ausgeben (von oben nach unten)
Buchstabe	1285	Variable buchstabe$ enthält die Buchstaben HERC.XPOS% sowie HERC.YPS% geben die Zielkoordinate an Kursiv.Mode% = 0 bewirkt Normalschrift Kursiv.Mode% = 1 bewirkt Kursivschrift
HercApertur	1343	Weil das Seitenverhältnis der Hercules-Grafikkarte nicht 1:1 ist, werden Kreise normalerweise wie Ellipsen dargestellt. Aus diesem Grund läßt sich mit der Variablen Herc.Apert die x-Achse stauchen, um vollkommen »runde« Kreise und Kreisbögen zu erhalten. Der Wert dieser Variablen wird auch beim Zeichnen von Linien berücksichtigt.
HercDrucken	1353	keine Parameter, erzeugt Hardcopy auf Epson-LQ-Drucker
HercSpiegel	1383	keine Parameter, erzeugt Spiegelbild für die Hardcopy
Reversal	1408	keine Parameter, spiegelt die Bits eines Bytes für die Routine HercSpiegel

Tabelle 3. Die Unterprogramme zum Zeichnen einfacher geometrischer Figuren und zur Ausgabe von Text

wird, muß der Wert 3 geschrieben werden. Damit wird erreicht, daß der Grafikmodus eingeschaltet und der Bereich des Bildschirmspeichers von Adresse B0000 bis BFFFFhex zur Verfügung steht.

- In das Steuerregister des Video-Controllers, das über die Portadresse 3B8hex angesprochen wird, muß der Wert 0Ahex geschrieben werden, falls die Speicherseite 0, also der Datenbereich von Adresse B0000 bis B7FFFhex, angezeigt werden soll, oder 8Ahex, falls die Speicherseite 1 (der Bereich von Adresse B8000 bis BFFFFhex) ausgewählt werden soll.
- Ferner muß mit der GW-Basic-Anweisung

```
def seg=&HB000
```

das Datensegment für Seite 0 oder mit

```
def seg=&HB800
```

das Datensegment für Seite 1 gesetzt werden.

- Anschließend werden an die Datenregister des Video-Controllers, das über die Portadresse 3B5hex angesprochen wird, nacheinander die Daten 35, 2D, 2E, 07, 5B, 02, 57, 57, 02, 03, 00 und 00 übertragen. Dadurch ist der Grafikmodus der Hercules-Karte eingeschaltet.

Für das Ausschalten des Grafikmodus ist eine ähnliche Vorgehensweise erforderlich, die dem Listing entnommen werden kann.

Übrigens empfiehlt es sich, die beiden Routinen zum Ein- und Ausschalten des Hercules-Grafikmodus auch als eigenständige Programme zu verwenden, weil man so in der Lage ist, den Grafikmodus beliebig ein- und auszuschalten. Gerade das Ausschalten des Hercules-Grafikmodus beziehungsweise das Einschalten des Textmodus ist manchmal wichtig, und zwar dann, wenn ein Programm verlassen wurde, das die Hercules-Karte nicht wieder in den Textmodus zurückgesetzt hat.

Soll ein Byte im Bildschirmspeicher der Hercules-Grafikkarte angesprochen werden, so muß als erstes dessen

Adresse nach folgender Formel berechnet werden:

```
adr%=8192*(Y.B% MOD 4)+90*(Y.B%\4)+(X.B%\8)
```

Die Variable X.B% (Wertebereich von 0 bis 89) stellt den x-Anteil der Byte-Koordinate dar, und Y.B% (Wertebereich von 0 bis 347) stellt den y-Anteil der Byte-Koordinate dar. In eine Bildzeile passen folglich 90 Byte zu je 8 Bildpunkten, was insgesamt 720 Bildpunkte pro Zeile ergibt. Insgesamt stellt die Hercules-Karte 348 Bildzeilen dar, was eine Auflösung von insgesamt 720x348 Bildpunkten zur Folge hat.

Zum Einschalten des Grafikmodus der Hercules-Karte sind mehrere Programmschritte erforderlich

Jedes Byte des Hercules-Bildschirmspeichers wird in der Toolbox über eine Koordinate angesprochen. Daher müssen vor dem Aufruf der Routinen, die ein Byte setzen oder löschen, die beiden Variablen X.B% und Y.B% gesetzt werden. Das Koordinatenpaar X.B% = 0, Y.B% = 0 gibt beispielsweise das Byte in der linken oberen Ecke an.
Mit der Anweisung

```
poke adr%,&HFF
```

wird zum Beispiel das Byte unter der Adresse »adr%« mit dem Wert FFhex geladen, so daß alle Bits des Bytes gesetzt sind. Dieses Byte erscheint dann auf dem Monitor als kurzer, heller Strich.
Soll ein einzelner Bildpunkt angesprochen werden, so muß zunächst das betreffende Byte im Bildschirmspeicher ermittelt werden, das für diesen Bildpunkt zuständig ist. Ein Bildpunkt wird durch eine x/y-Koordinate definiert. Die Variable XP.P% (Wertebereich von 0 bis 719) stellt dabei den x-Anteil dar, und YP.P% (Wertebereich von 0 bis 347) gibt entsprechend den y-Anteil an.
Der Punkt für XP.P% = 0 und YP.P% = 0 liegt beispielsweise in der linken oberen Ecke des Bildschirms. Dies gilt zumindest für die elementare Routine »PunktSet« oder für die anderen Routinen im dritten Programmteil.
Vom Anwender wird allerdings die Routine »HercPktSet« im zweiten Programmteil aufgerufen. Über eine Koordinatentransformation wird dafür gesorgt, daß bei diesem Aufruf die Koordinate 0,0 in der linken unteren Ecke des Bildschirms liegt, der Bildschirm also den ersten Quadranten eines rechtwinkligen Koordinatensystems darstellt.
Mit Hilfe der Variablen X.NULL% und Y.NULL% in der Routine »Check« läßt sich übrigens eine Koordinatenverschiebung durchführen. Vor Aufruf der Routine »Punkt« sind in diesem Fall die Variablen X.P% für den x-Anteil der Koordinate (Wertebereich von 0 bis 719) und Y.P% für den y-Anteil (Wertebereich von 0 bis 347) zu setzen.
Der Aufbau der Elementarroutine zum Setzen eines Bildpunktes ist folgendermaßen:

- Zuerst wird das Byte im Bildschirmspeicher der Hercules-Karte bestimmt, in dem der Bildpunkt gesetzt werden soll.
- Damit sichergestellt ist, daß nicht aufgrund falscher Koordinatenangaben im Arbeitsspeicher statt Bildschirmspeicher Veränderungen vorgenommen werden, wird zunächst der zulässige Bereich für den zu setzenden oder zu löschenden Bildpunkt in der Routine »Check« überprüft. Werden die angegebenen Grenzen überschritten, so wird der maximal zulässige Grenzwert eingesetzt.
- Anschließend wird das betreffende Byte ausgelesen und das hell zu setzende Bit in dem Byte auf 1 gesetzt. Soll der Bildpunkt dagegen dunkel gesetzt, das heißt gelöscht werden, so wird das entsprechende Bit auf 0 gesetzt.

Vor dem ersten Aufruf der Grafikroutinen in der Hercules-Toolbox ist eine Initialisierung erforderlich, bei der wichtige Konstanten gesetzt werden und das Alphabet für die Buchstabendarstellung geladen wird. Die Buchstaben werden dabei in einer Matrix, bestehend aus 8x6 Bildpunkten, dargestellt.
Beim Zeichnen von Linien wird zunächst geprüft, ob nur ein einzelner Punkt, eine völlig horizontale oder eine vertikale Linie gewünscht ist. Ist dies der Fall, so wird der Bildpunkt beziehungsweise die Linie gezeichnet und die Routine beendet.
Soll dagegen eine schräge Linie gezeichnet werden, so bestimmt das Programm zuerst deren Steigung. Ist diese kleiner als 1, so wird die Linie mit der Schrittweite 1 längs der x-Achse gezeichnet. Dieses Verfahren hat den Vorteil, daß die zugehörigen y-Werte sich voneinander um nicht mehr als den Wert 1 unterscheiden und daher ein geschlossener Linienzug entsteht.

Die Hercules-Toolbox enthält auch Funktionen zum Zeichnen von Linien und einfachen geometrischen Figuren

Ist die Steigung der Linie größer als 1, so wird die Linie mit der Schrittweite 1 längs der y-Achse gezeichnet. Somit entsteht auch für diese Steigung ein geschlossener Linienzug.
Die Programmierung der Kreisroutine erfolgt in zwei Schritten. Im ersten Schritt wird der Teil von 0 bis 45 Grad bearbeitet und im zweiten der Teil von 45 bis 90 Grad. Durch Spiegelung werden die restlichen Bildpunkte gewonnen.
Bei der Kreisberechnung ist darauf zu achten, daß der Kreisumfang vollständig mit Punkten gefüllt ist. Es ist daher nicht sinnvoll, die Berechnung mit einer For-Next-Schleife über den Umfangswinkel zu steuern, weil der Umfang noch durch den Radius beeinflußt wird und es daher bei einer zu groben Wahl des Winkelinkrementes zu Lücken kommen könnte.
In der Hercules-Toolbox wurde daher folgender Weg gewählt: Ausgehend von 90 Grad nach 45 Grad, also vom obersten Punkt des Kreises nach rechts, wird parallel zur x-Achse ein Schritt nach rechts gegangen und die y-Position berechnet, die sich um nicht mehr als einen Schritt von der vorhergehenden unterscheidet. Dieser Bildpunkt und die durch Spiegelung gefundenen drei weiteren Punkte werden gesetzt, und anschließend wird der nächste Punkt bestimmt.
Sind die ersten 45 Grad eines Kreises bearbeitet, so wird der Kreis bei 90 Grad, das heißt ganz rechts, begonnen und nach dem gleichen Verfahren in Richtung 0 Grad gezeichnet. Auf dem Monitor sehen Sie dabei den Kreis von verschiedenen Punkten aus »zusammenwachsen«.
Die Programmierung eines Kreisbogens geht von der Ellipsengleichung aus. Der Anwender muß daher die beiden Achsen, den Mittelpunkt sowie den Start- und Endwinkel für den Kreisbogen angeben. Weil hier die Bogenlinie über den Winkel gesteuert wird, ist auch das Winkelinkrement anzugeben, um so schrittweise alle Koordinatenpunkte berechnen zu können. Entstehen auf der Bogenlinie Lücken, so muß das Winkelinkrement entsprechend verringert werden.
Bei der Entwicklung der Hercules-Toolbox wurde darauf Wert gelegt, daß sie sich in allen drei Basic-Dialekten (Quick Basic, SBasic und Turbo Basic) ohne Änderung verwenden läßt. Einzige Ausnahme: Weil in SBasic der Befehl »width« mit Parameter nicht akzeptiert wird, ist in der Routine »HercDrucken« in Programmzeile 1630 nur der Befehl allein enthalten. Arbeiten Sie dagegen mit Quick Basic oder Turbo Basic, so muß diese Programmzeile durch ein führendes »rem« zur Kommentarzeile erweitert und in der vorhergehenden Zeile 1359 der Rem-Befehl entfernt werden. (Dr. H.D. Gellißen/ma)

Name	hercules.bas
Sprache	Quick Basic, SBasic, Turbo Basic
Grafikkarte	nur Hercules-Grafikkarte
Besonderheiten	Routinen-Sammlung zur Programmierung der Hercules-Grafikkarte

```
[8C4]    1: '----------------------------------------
[3R5]    2: 'Hercules-Grafik für Quick Basic,
[GK4]    3: '          SBasic und Turbo Basic
[4W6]    4: '(C) 1989 Redaktionsbüro Everts&Hagedorn
[KW3]    5: '          DOS International
[6U4]    6: '          Autor: Dr. H.D. Gellißen
[SB6]    7: 'Achtung: In Zeile 1359 ist für SBasic
[135]    8: '          der erste WIDTH-Befehl als
[VG5]    9: '          Kommentar gesetzt. Dies ist
[SS5]   1Ø: '          nötig, weil SBasic »WIDTH«
[BG5]   11: '          nur ohne Parameter versteht.
[8C4]   12: '----------------------------------------
[AA2]   13: GOSUB Herc.Ini
[6K5]   14: GOSUB HercEin: GOSUB HercLoeschen
[K82]   15: GOSUB HercUhr
[Y81]   16: Anfang:
[PNØ]   17: CLS
[GXØ]   18: PRINT
[KY5]   19: PRINT "       HERCULES - TESTPROGRAMME"
[GXØ]   2Ø: PRINT
[ØØØ]   21:
[E99]   22: PRINT "Testende:                  Ø     Hercu
            les-Texteinstellung: 1"
[C18]   23: PRINT "Videoseite Ø:              2     Video
            seite 1:              3"
[WT4]   24: PRINT "Schreibseite Ø/1:          4 "
[TA9]   25: PRINT "Bildschirm hell:           5     Bilds
            chirm löschen:        6"
[UA8]   26: PRINT "Byte setzen:               7     Byte
            löschen:              8"
[3Z8]   27: PRINT "Punkt setzen:              9     Punkt
             löschen:             1Ø"
[M37]   28: PRINT "Kreis:                     11    Linie
            :                     12"
[DH7]   29: PRINT "Rechteck:                  13    Dreie
            ck:                   14"
[CS7]   3Ø: PRINT "Kreisbogen:                15    Winke
            l:                    16"
[M89]   31: PRINT "Bildseiten (A,O,X):        17    Bilds
            eiten toggeln:        18"
[DE9]   32: PRINT "Grafik laden/speichern:    19    Grafi
            k drucken:            2Ø"
[369]   33: PRINT "Grafik beschriften:        21    Grafi
            k spiegeln:           22"
[UV9]   34: PRINT "Aperturfaktor einstellen:23     Uhr a
            nzeigen:               24"
[GXØ]   35: PRINT
[4GE]   36: PRINT "Seite "; Herc.Video.Flag%; " auf Mon
            itor, Seite "; Herc.Seiten.Flag%; " schreib
            aktiv."
[9M5]   37: INPUT "Bitte Zahl eingeben: ", zahl%
[ØØØ]   38:
[VY8]   39: IF zahl% = Ø THEN PRINT "ENDE DES HERCULES-
            TESTES!": STOP
[YØ7]   4Ø: IF zahl% = 1 THEN GOSUB HercAus: GOTO Anfan
            g
[8M7]   41: IF zahl% = 2 THEN GOSUB HercZeigeØ: GOTO An
            fang
[CM7]   42: IF zahl% = 3 THEN GOSUB HercZeigel: GOTO An
            fang
[718]   43: IF zahl% = 4 THEN GOSUB HercSchreiben: GOTO
             Anfang
[4Z6]   44: IF zahl% = 5 THEN GOSUB HercHel: GOTO Anfan
            g
[EP7]   45: IF zahl% = 6 THEN GOSUB HercLoesch: GOTO An
            fang
[WH8]   46: IF zahl% = 7 THEN GOSUB HercByteSetzen: GOT
            O Anfang
[EG8]   47: IF zahl% = 8 THEN GOSUB HercByteLoeschen: G
            OTO Anfang
[HT7]   48: IF zahl% = 9 THEN GOSUB HercPktSet: GOTO An
            fang
[Y98]   49: IF zahl% = 1Ø THEN GOSUB HercPktLoesch: GOT
            O Anfang
[ZL7]   5Ø: IF zahl% = 11 THEN GOSUB HercKreis: GOTO An
            fang
[ZH7]   51: IF zahl% = 12 THEN GOSUB HercLinie: GOTO An
            fang
[6Y7]   52: IF zahl% = 13 THEN GOSUB HercRechteck: GOTO
             Anfang
[NK7]   53: IF zahl% = 14 THEN GOSUB HercDreieck: GOTO
            Anfang
[BK7]   54: IF zahl% = 15 THEN GOSUB HercBogen: GOTO An
            fang
[EL7]   55: IF zahl% = 16 THEN GOSUB HercWinkel: GOTO A
            nfang
[9L7]   56: IF zahl% = 17 THEN GOSUB HercSeiten: GOTO A
            nfang
[KL7]   57: IF zahl% = 18 THEN GOSUB HercToggel: GOTO A
            nfang
[2B8]   58: IF zahl% = 19 THEN GOSUB HercSpeichern: GOT
            O Anfang
[3M7]   59: IF zahl% = 2Ø THEN GOSUB HercDrucken:GOTO A
            nfang
[D87]   6Ø: IF zahl% = 21 THEN GOSUB HercText: GOTO Anf
            ang
[2M7]   61: IF zahl% = 22 THEN GOSUB HercSpiegel: GOTO
            Anfang
[HP7]   62: IF zahl% = 23 THEN GOSUB HercApertur: GOTO
            Anfang
[PX6]   63: IF zahl% = 24 THEN GOSUB HercUhr: GOTO Anfa
            ng
[312]   64: GOTO Anfang
[ØØØ]   65:
[8C4]   66: '----------------------------------------
[EX6]   67: 'Es folgen Beispiele, um die Fähigkeiten
[NP6]   68: ' der Hercules-Grafikroutinen zu zeigen
[8C4]   69: '----------------------------------------
[ØØØ]   7Ø:
[8C4]   71: '----------------------------------------
[6W5]   72: 'HercZeigeØ: Zeigt Grafik auf Seite Ø
[4Ø5]   73: ' bis eine Taste gedrückt wird
[8C4]   74: '----------------------------------------
[UY1]   75: HercZeigeØ:
[8P3]   76:    Herc.Seite.Video% = &HA
[9K3]   77:    Herc.Seg.Video% = &HBØØØ
[833]   78:    Herc.Video.Flag% = Ø
[X17]   79:    GOSUB HercEin: GOSUB Warten: GOSUB HercA
            us
[651]   8Ø:  RETURN
[ØØØ]   81:
[8C4]   82: '----------------------------------------
[CW5]   83: 'HercZeigel: Zeigt Grafik auf Seite 1
[4Ø5]   84: ' bis eine Taste gedrückt wird
[8C4]   85: '----------------------------------------
[XY1]   86: HercZeigel:
[5R3]   87:    Herc.Seite.Video% = &H8A
[1L3]   88:    Herc.Seg.Video% = &HB8ØØ
[933]   89:    Herc.Video.Flag% = 1
[X17]   9Ø:    GOSUB HercEin: GOSUB Warten: GOSUB HercA
            us
[651]   91:  RETURN
[ØØØ]   92:
[8C4]   93: '----------------------------------------
[KY5]   94: 'HercSchreiben: Legt die Seite fest,
[R14]   95: ' auf die geschrieben wird
[W75]   96: '&HBØØØ = Seite Ø und &HB8ØØ = Seite 1
[8C4]   97: '----------------------------------------
[7E2]   98: HercSchreiben:
[U84]   99:    Herc.Seg.Schreiben% = &HB8ØØ
[MA3]  1ØØ:    Herc.Seiten.Flag% = 1
[7X8]  1Ø1:    INPUT "Welche Seite ( Ø oder 1 ) wird be
            schrieben: ", zahl%
[M8A]  1Ø2:    IF zahl% = Ø! THEN Herc.Seg.Schreiben% =
             &HBØØØ: Herc.Seiten.Flag% = Ø
[651]  1Ø3: RETURN
[ØØØ]  1Ø4:
[8C4]  1Ø5: '----------------------------------------
[X36]  1Ø6: 'HercLoeschen: Löscht den Bildschirm
[NB5]  1Ø7: ' durch Setzen der Bytes auf ØØhex
[8C4]  1Ø8: '----------------------------------------
[WØ2]  1Ø9: HercLoesch:
[P28]  11Ø:    GOSUB HercEin: GOSUB HercLoeschen: GOSUB
             HercAus
[651]  111:  RETURN
[ØØØ]  112:
[8C4]  113: '----------------------------------------
[C46]  114: 'HercHel: Schaltet den Bildschirm hell
[FG5]  115: ' durch Setzen der Bytes auf FFhex
[8C4]  116: '----------------------------------------
[LD1]  117: HercHel:
[HA7]  118:    GOSUB HercEin: GOSUB HercHell: GOSUB Her
            cAus
[651]  119:  RETURN
[ØØØ]  12Ø:
[ØØØ]  121:
[8C4]  122: '----------------------------------------
[WR6]  123: 'HercByteSetzen: Setzt das Byte auf der
[9D3]  124: ' Koordinate x.b%, y.b%
[F26]  125: 'x.b%=Ø und y.b%=Ø bezeichnet die linke
[4X1]  126: ' obere Ecke
[8C4]  127: '----------------------------------------
[9W2]  128: HercByteSetzen:
[WC8]  129:   PRINT "Byte in Spalte x und Zeile z wird
            gelöscht"
[Y73]  13Ø:   INPUT "x-Byte :", x.b%
[383]  131:   INPUT "y-Byte :", y.b%
[L6A]  132:   GOSUB HercEin: GOSUB SetzeByte: GOSUB War
            ten: GOSUB HercAus
[651]  133:  RETURN
[ØØØ]  134:
[8C4]  135: '----------------------------------------
[WA6]  136: 'HercByteLoeschen: Löscht das Byte auf
[2Ø4]  137: ' der Koordinate x.b%, y.b%
[F26]  138: 'x.b%=Ø und y.b%=Ø bezeichnet die linke
[4X1]  139: ' obere Ecke
[8C4]  14Ø: '----------------------------------------
```

```
[E13]  141: HercByteLoeschen:
[WC8]  142:   PRINT "Byte in Spalte x und Zeile z wird
       gelöscht"
[Y73]  143:   INPUT "x-Byte :", x.b%
[383]  144:   INPUT "y-Byte :", y.b%
[GBA]  145:    GOSUB HercEin: GOSUB LoescheByte: GOSUB
       Warten: GOSUB HercAus
[651]  146:  RETURN
[ØØØ]  147:
[8C4]  148: '---------------------------------------
[XZ5]  149: 'HercPktSet: Setzt den Punkt auf der
[1F3]  15Ø: ' Koordinate x.p%, y.p%
[N46]  151: 'x.p%=Ø und y.p%=Ø bezeichnet die linke
[422]  152: ' untere Ecke
[8C4]  153: '---------------------------------------
[W32]  154: HercPktSet:
[D68]  155:   PRINT "Punkt auf der x,y-Koordinate wird
       gesetzt"
[WC4]  156:   INPUT "x-Koordinate :", x.p%
[ZC4]  157:   INPUT "y-Koordinate :", y.p%
[VH9]  158:   GOSUB HercEin: GOSUB Punkt: GOSUB Warten:
       GOSUB HercAus
[651]  159:  RETURN
[ØØØ]  16Ø:
[8C4]  161: '---------------------------------------
[XV6]  162: 'HercPktLoesch: Löscht den Punkt auf der
[1F3]  163: ' Koordinate x.p%, y.p%
[N46]  164: 'x.p%=Ø und y.p%=Ø bezeichnet die linke
[422]  165: ' untere Ecke
[8C4]  166: '---------------------------------------
[1K2]  167: HercPktLoesch:
[DE8]  168:   PRINT "Punkt auf der x,y-Koordinate wird
       gelöscht"
[WC4]  169:   INPUT "x-Koordinate :", x.p%
[ZC4]  17Ø:   INPUT "y-Koordinate :", y.p%
[HE9]  171:   GOSUB HercEin: GOSUB LPunkt: GOSUB Warten
       : GOSUB HercAus
[651]  172:  RETURN
[ØØØ]  173:
[8C4]  174: '---------------------------------------
[8N5]  175: 'HercKreis: Zeichnet Kreis mit dem
[LA5]  176: ' Mittelpunkt auf der Koordinate
[RH4]  177: ' x.k%,y.k% und dem Radius x.r
[8C4]  178: '---------------------------------------
[7T1]  179: HercKreis:
[X3D]  18Ø:   PRINT "Kreis zeichnen oder löschen x,y -
       Mittelpunktskoordinaten, x.r - Radius"
[MH5]  181:   INPUT "Neue Koordinaten (j/n)", in$
[8P6]  182:   IF in$ = "j" OR in$ = "J" THEN GOTO HercK
       reis1
[ZP2]  183:   GOTO HercKreis2
[MZ1]  184: HercKreis1:
[8C4]  185:    INPUT "x-Koordinate :", x.k%
[BC4]  186:    INPUT "y-Koordinate :", y.k%
[CP3]  187:    INPUT "Radius       :", x.r
[RZ1]  188: HercKreis2:
[VN8]  189:  INPUT "Figur löschen=1, XOR=x, sonst zeich
       nen ", in$
[552]  19Ø:    Modepunkt% = 1
[U66]  191:    IF in$ = "1" OR in$ = "L" THEN Modepunkt
       % = 2
[S86]  192:    IF in$ = "x" OR in$ = "X" THEN Modepunkt
       % = 3
[GG9]  193:    GOSUB HercEin: GOSUB Kreis: GOSUB Warten
       : GOSUB HercAus
[651]  194:  RETURN
[ØØØ]  195:
[8C4]  196: '---------------------------------------
[6E6]  197: 'HercLinie: Zeichnet oder löscht Linie
[CP4]  198: ' von x.11%,y.11% bis x.12%,y.12%
[8C4]  199: '---------------------------------------
[PS1]  2ØØ: HercLinie:
[KW5]  2Ø1:   PRINT "Linie zeichnen oder löschen"
[XL5]  2Ø2:   INPUT "Neue Koordinaten (j/n) ", in$
[RN6]  2Ø3:   IF in$ = "j" OR in$ = "J" THEN GOTO HercL
       inie1
[XM2]  2Ø4:   GOTO HercLinie2
[3Z1]  2Ø5: HercLinie1:
[BP3]  2Ø6:     INPUT "x-Anfang :", x.11%
[EP3]  2Ø7:     INPUT "y-Anfang :", y.11%
[SD3]  2Ø8:     INPUT "x-Ende   :", x.12%
[VD3]  2Ø9:     INPUT "y-Ende   :", y.12%
[6Z1]  21Ø: HercLinie2:
[VN8]  211:   INPUT "Figur löschen=1, XOR=x, sonst zeic
       hnen ", in$
[552]  212:    Modepunkt% = 1
[U66]  213:    IF in$ = "1" OR in$ = "L" THEN Modepunkt
       % = 2
[S86]  214:    IF in$ = "x" OR in$ = "X" THEN Modepunkt
       % = 3
[EE9]  215:   GOSUB HercEin: GOSUB Linie: GOSUB Warten:
       GOSUB HercAus
[651]  216:  RETURN
[ØØØ]  217:
[8C4]  218: '---------------------------------------
[DA6]  219: 'HercRechteck: Zeichnet oder löscht
[NM1]  22Ø: ' Rechteck
[NF6]  221: 'x1,y1= unten links, x2,y2= oben rechts
[8C4]  222: '---------------------------------------
[L72]  223: HercRechteck:
[YDC]  224:   PRINT "Rechteck zeichnen oder löschen: x1
       ,y1 unten links, x2,y2 oben rechts"
[XL5]  225:   INPUT "Neue Koordinaten (j/n) ", in$
[X47]  226:   IF in$ = "j" OR in$ = "J" THEN GOTO HercR
       echteck1
[U33]  227:   GOTO HercRechteck2
[FC2]  228: HercRechteck1:
[6G9]  229:    PRINT "Box zeichnen: x1,y1 - unten link
       s, x2,y2 - oben rechts"
[3N3]  23Ø:     INPUT "x-Anfang :", x.b1%
[6N3]  231:     INPUT "y-Anfang :", y.b1%
[GC3]  232:     INPUT "x-Ende   :", x.b2%
[LC3]  233:     INPUT "y-Ende   :", y.b2%
[HC2]  234: HercRechteck2:
[VN8]  235:   INPUT "Figur löschen=1, XOR=x, sonst zeic
       hnen ", in$
[552]  236:    Modepunkt% = 1
[U66]  237:    IF in$ = "1" OR in$ = "L" THEN Modepunkt
       % = 2
[S86]  238:    IF in$ = "x" OR in$ = "X" THEN Modepunkt
       % = 3
```

Listing. Die Routinensammlung »hercules.bas« läßt sich mit Quick Basic, SBasic und Turbo Basic compilieren

Software-Autoren gesucht

Haben Sie interessante Programme geschrieben?

Zum Beispiel Anwendungen, die zu umfangreich sind, um als Listing abgedruckt zu werden? Oder Programmiertools, die mit professionellen Produkten konkurrieren können? Spiele? Grafikprogramme? Datenbanken? Utilities?

Haben Sie Ideen für neue Software?

Vielleicht haben Sie bereits begonnen zu programmieren, aber scheitern an einem bestimmten Problem?
Oder Sie können aus Zeitmangel nicht alle Ideen realisieren?

Haben Sie Erfahrung als Programmierer?

Suchen Sie eine Aufgabe, die fasziniert? Wollen Sie im Team ein Projekt realisieren?

Zögern Sie nicht, uns Ihre Programme und Ideen zuzusenden. Gern werden wir uns gegebenenfalls mit Ihnen um eine Vermarktung der Produkte kümmern. Wenn Sie interessiert sind, schicken Sie uns die aktuelle Version des Programms auf Datenträger und einen Ausdruck der Dokumentation.

Redaktionsbüro Everts & Hagedorn,
Redaktion DOS International, Stichwort »Software-Autor«, Hans-Stießberger-Str. 3, 8013 Haar

```
[ØØØ]  239:
[B62]  24Ø:   GOSUB HercEin
[PT9]  241:   x.l1% = x.b1%: y.l1% = y.b1%: x.l2% = x.b
       2%: y.l2% = y.b1%: GOSUB Linie
[TT9]  242:   x.l1% = x.b2%: y.l1% = y.b1%: x.l2% = x.b
       2%: y.l2% = y.b2%: GOSUB Linie
[UT9]  243:   x.l1% = x.b2%: y.l1% = y.b2%: x.l2% = x.b
       1%: y.l2% = y.b2%: GOSUB Linie
[RT9]  244:   x.l1% = x.b1%: y.l1% = y.b2%: x.l2% = x.b
       1%: y.l2% = y.b1%: GOSUB Linie
[2V4]  245:   GOSUB Warten: GOSUB HercAus
[ØØØ]  246:
[651]  247:  RETURN
[ØØØ]  248:
[8C4]  249: '---------------------------------------
[ZZ5]  25Ø: 'HercDreieck: Zeichnet oder löscht
[VX1]  251: ' ein Dreieck
[6T5]  252: 'x1,y1, x2,y2 und x3,y3 sind die Ecken
[8C4]  253: '---------------------------------------
[Y32]  254: HercDreieck:
[XL5]  255:   INPUT "Neue Koordinaten (j/n) ", in$
[H17]  256:   IF in$ = "j" OR in$ = "J" THEN GOTO HercD
       reieck1
[LW2]  257:   GOTO HercDreieck2
[972]  258: HercDreieck1:
[FT9]  259:     PRINT "Dreieck zeichnen: x1,y1, x2,y2 u
       nd x3,y3 sind die Ecken"
[UN2]  26Ø:     INPUT "x1 :", x.d1%
[ZN2]  261:     INPUT "y1 :", y.d1%
[ZN2]  262:     INPUT "x2 :", x.d2%
[4P2]  263:     INPUT "y2 :", y.d2%
[4P2]  264:     INPUT "x3 :", x.d3%
[9P2]  265:     INPUT "y3 :", y.d3%
[A72]  266: HercDreieck2:
[VN8]  267:   INPUT "Figur löschen=1, XOR=x, sonst zeic
       hnen ", in$
[552]  268:    Modepunkt% = 1
[U66]  269:    IF in$ = "l" OR in$ = "L" THEN Modepunkt
       % = 2
[S86]  27Ø:    IF in$ = "x" OR in$ = "X" THEN Modepunkt
       % = 3
[ØØØ]  271:
[B62]  272:   GOSUB HercEin
[CU9]  273:   x.l1% = x.d1%: y.l1% = y.d1%: x.l2% = x.d
       2%: y.l2% = y.d2%: GOSUB Linie
[PU9]  274:   x.l1% = x.d2%: y.l1% = y.d2%: x.l2% = x.d
       3%: y.l2% = y.d3%: GOSUB Linie
[HU9]  275:   x.l1% = x.d3%: y.l1% = y.d3%: x.l2% = x.d
       1%: y.l2% = y.d1%: GOSUB Linie
[2V4]  276:   GOSUB Warten: GOSUB HercAus
[651]  277:  RETURN
[8C4]  278: '---------------------------------------
[UX6]  279: 'HercBogen: Zeichnet Bogen um Mittelpunkt
[7K6]  28Ø: ' x.k%,y.k% mit Radius x.r von Winkel α
[416]  281: ' bis β in Grad-Angabe (d.h. Ø bis 36Ø)
[8C4]  282: '---------------------------------------
[6S1]  283: HercBogen:
[S8C]  284:   PRINT "Kreisbogen zeichnen oder löschen:
       x,y - Mittelpunktskoordinaten"
[MH5]  285:   INPUT "Neue Koordinaten (j/n)", in$
[7N6]  286:   IF in$ = "j" OR in$ = "J" THEN GOTO HercB
       ogen1
[3N2]  287:   GOTO HercBogen2
[LY1]  288: HercBogen1:
[AD4]  289:    INPUT "x-Mittelpunkt :", x.k
[ED4]  29Ø:    INPUT "y-Mittelpunkt :", y.k
[HC4]  291:    INPUT "x-Achse       :", x.Achse
[NC4]  292:    INPUT "y-Achse       :", y.Achse
[8M7]  293:    INPUT "Von Winkel alpha - Angabe in Grad
       :", alpha
[X77]  294:    INPUT "Bis Winkel beta  - Angabe in Grad
       :", beta
[N59]  295:    INPUT "Winkelinkrement in Grad oder Dezm
       ale davon:", dwi
[5U2]  296:    HercBogenMode% = Ø
[UD9]  297:    INPUT " Segment zeichnen=1, sonst Bogen
       :", HercBogenMode%
[PY1]  298: HercBogen2:
[VN8]  299:  INPUT "Figur löschen=1, XOR=x, sonst zeich
       nen ", in$
[552]  3ØØ:    Modepunkt% = 1
[U66]  3Ø1:    IF in$ = "l" OR in$ = "L" THEN Modepunkt
       % = 2
[S86]  3Ø2:    IF in$ = "x" OR in$ = "X" THEN Modepunkt
       % = 3
[ME9]  3Ø3:    GOSUB HercEin: GOSUB Bogen: GOSUB Warten
       : GOSUB HercAus
[651]  3Ø4:  RETURN
[ØØØ]  3Ø5:
[8C4]  3Ø6: '---------------------------------------
[VX6]  3Ø7: 'HercWinkel: Zeichnet oder löscht Winkel
[6T5]  3Ø8: 'x1,y1, x2,y2 und x3,y3 sind die Ecken
[8C4]  3Ø9: '---------------------------------------
[C22]  31Ø: HercWinkel:
[XL5]  311:   INPUT "Neue Koordinaten (j/n) ", in$
[4Z6]  312:   IF in$ = "j" OR in$ = "J" THEN GOTO HercW
       inkel1
[WU2]  313:   GOTO HercWinkel2
[232]  314: HercWinkel1:
[VW8]  315:     PRINT "Winkel zeichnen: x1,y1 - x2,y2 -
       x3,y3 sind die Ecken"
[UN2]  316:     INPUT "x1 :", x.d1%
[ZN2]  317:     INPUT "y1 :", y.d1%
[ZN2]  318:     INPUT "x2 :", x.d2%
[4P2]  319:     INPUT "y2 :", y.d2%
[4P2]  32Ø:     INPUT "x3 :", x.d3%
[9P2]  321:     INPUT "y3 :", y.d3%
[632]  322: HercWinkel2:
[VN8]  323:   INPUT "Figur löschen=1, XOR=x, sonst zeic
       hnen ", in$
[552]  324:    Modepunkt% = 1
[U66]  325:    IF in$ = "l" OR in$ = "L" THEN Modepunkt
       % = 2
[S86]  326:    IF in$ = "x" OR in$ = "X" THEN Modepunkt
       % = 3
[ØØØ]  327:
[B62]  328:   GOSUB HercEin
[CU9]  329:   x.l1% = x.d1%: y.l1% = y.d1%: x.l2% = x.d
       2%: y.l2% = y.d2%: GOSUB Linie
[PU9]  33Ø:   x.l1% = x.d2%: y.l1% = y.d2%: x.l2% = x.d
       3%: y.l2% = y.d3%: GOSUB Linie
[2V4]  331:  GOSUB Warten: GOSUB HercAus
[ØØØ]  332:
[651]  333:  RETURN
[ØØØ]  334:
[8C4]  335: '---------------------------------------
[V67]  336: ' HercSeiten: Verknüpft zwei Grafikseiten
[8C4]  337: '---------------------------------------
[L12]  338: HercSeiten:
[457]  339: INPUT "AND=Ø, OR=1, XOR=2 :", Herc.Seiten.m
       ode%
[HZ8]  34Ø:    Herc.Seite.Video% = &H8A 'Es wird die er
       ste Seite gezeigt
[1L3]  341:    Herc.Seg.Video% = &HB8ØØ
[933]  342:    Herc.Video.Flag% = 1
[ØØØ]  343:
[KCA]  344:   GOSUB HercEin: GOSUB HercBildMode: GOSUB
       Warten: GOSUB HercAus
[ØØØ]  345:
[651]  346: RETURN
[ØØØ]  347:
[8C4]  348: '---------------------------------------
[KT5]  349: 'HercToggel: Schaltet zwischen den
[UU5]  35Ø: ' Bildschirmseiten Ø und 1 hin und her
[8C4]  351: '---------------------------------------
[S12]  352: HercToggel:
[A4C]  353: INPUT "Zeit zwischen Bildwechsel angeben mi
       t einer Zahl (>1ØØ)", Herc.Delay
[FB2]  354: Toggelweiter:
[232]  355:  GOSUB Toggel
[EL1]  356:  a$ = INKEY$
[H35]  357:  IF a$ = "" THEN GOTO Toggelweiter
[H62]  358:  GOSUB HercAus
[651]  359: RETURN
[ØØØ]  36Ø:
[8C4]  361: '---------------------------------------
[TM4]  362: 'HercText: Gibt Text auf der
[NY2]  363: ' Schreibseite aus
[8C4]  364: '---------------------------------------
[9M1]  365: HercText:
[3X8]  366:  INPUT "Neuen Text eingeben = 1, sonst alte
       r Text ", hilf%
[2X5]  367:  IF hilf% = Ø THEN GOTO HercTextWeiter
[XC7]  368:  LINE INPUT "Bitte Text eingeben: ", Herc.s
       atz$
[ØZ7]  369:  INPUT "An welcher x-Position: ", Herc.xSat
       z.Pos%
[8Z7]  37Ø:  INPUT "An welcher y-Position: ", Herc.ySat
       z.Pos%
[E89]  371:  INPUT "Horizontal = Ø oder vertikal = 1? "
       , herc.Satz.Mode%
[XC7]  372:  INPUT "Kursiv =1, sonst steil: ", kursiv.m
       ode%
[KV2]  373: HercTextWeiter:
[HH8]  374:  INPUT "Text löschen=1, XOR=x, sonst zeichn
       en ", in$
[552]  375:    Modepunkt% = 1
[U66]  376:    IF in$ = "l" OR in$ = "L" THEN Modepunkt
       % = 2
[S86]  377:    IF in$ = "x" OR in$ = "X" THEN Modepunkt
       % = 3
[ØØØ]  378:
[6C7]  379:  GOSUB HercEin: GOSUB HercSatz: GOSUB HercA
       us
[651]  38Ø:  RETURN
[ØØØ]  381:
[8C4]  382: '---------------------------------------
[W96]  383: 'HercUhr: Zeigt die Uhr auf Seite 1 an
[8C4]  384: '---------------------------------------
[EF1]  385: HercUhr:
[ØØØ]  386:
[7A1]  387: hilf$ = ""
[R91]  388: x.k% = 225
[F91]  389: y.k% = 15Ø
[G61]  39Ø: x.r = 1ØØ
[552]  391: Modepunkt% = 1
```

```
[6D4]   392: GOSUB HercEin: GOSUB Kreis
[7DC]   393: herc.xpos% = 219 * Herc.Apert: herc.ypos% =
             253: buchstabe$ = "1": GOSUB buchstabe
[9DC]   394: herc.xpos% = 226 * Herc.Apert: herc.ypos% =
             253: buchstabe$ = "2": GOSUB buchstabe
[DDC]   395: herc.xpos% = 272 * Herc.Apert: herc.ypos% =
             241: buchstabe$ = "1": GOSUB buchstabe
[9EC]   396: herc.xpos% = 315 * Herc.Apert: herc.ypos% =
             199: buchstabe$ = "2": GOSUB buchstabe
[PDC]   397: herc.xpos% = 328 * Herc.Apert: herc.ypos% =
             146: buchstabe$ = "3": GOSUB buchstabe
[V6C]   398: herc.xpos% = 315 * Herc.Apert: herc.ypos% =
              96: buchstabe$ = "4": GOSUB buchstabe
[Y6C]   399: herc.xpos% = 273 * Herc.Apert: herc.ypos% =
              52: buchstabe$ = "5": GOSUB buchstabe
[66C]   4ØØ: herc.xpos% = 223 * Herc.Apert: herc.ypos% =
              4Ø: buchstabe$ = "6": GOSUB buchstabe
[27C]   4Ø1: herc.xpos% = 172 * Herc.Apert: herc.ypos% =
              52: buchstabe$ = "7": GOSUB buchstabe
[97C]   4Ø2: herc.xpos% = 131 * Herc.Apert: herc.ypos% =
              96: buchstabe$ = "8": GOSUB buchstabe
[LDC]   4Ø3: herc.xpos% = 119 * Herc.Apert: herc.ypos% =
             146: buchstabe$ = "9": GOSUB buchstabe
[5EC]   4Ø4: herc.xpos% = 131 * Herc.Apert: herc.ypos% =
             199: buchstabe$ = "Ø": GOSUB buchstabe
[6EC]   4Ø5: herc.xpos% = 125 * Herc.Apert: herc.ypos% =
             199: buchstabe$ = "1": GOSUB buchstabe
[ADC]   4Ø6: herc.xpos% = 172 * Herc.Apert: herc.ypos% =
             241: buchstabe$ = "1": GOSUB buchstabe
[9DC]   4Ø7: herc.xpos% = 165 * Herc.Apert: herc.ypos% =
             241: buchstabe$ = "1": GOSUB buchstabe
[ØØØ]   4Ø8:
[263]   4Ø9: Herc.xSatz.Pos% = 5Ø
[Z83]   41Ø: Herc.ySatz.Pos% = 32Ø
[8CC]   411: Herc.satz$ = "Die Hercules-Uhr von DOS Inte
             rnational in Basic programmiert:"
[552]   412: Modepunkt% = 1
[LG2]   413: kursiv.mode% = Ø
[NG2]   414: GOSUB HercSatz
[ØØØ]   415:
[263]   416: Herc.xSatz.Pos% = 5Ø
[N53]   417: Herc.ySatz.Pos% = 1Ø
[L38]   418: Herc.satz$ = "Zum Fortfahren eine Taste dru
             ecken!"
[552]   419: Modepunkt% = 1
[LG2]   42Ø: kursiv.mode% = Ø
[NG2]   421: GOSUB HercSatz
[ØØØ]   422:
[7A1]   423: hilf$ = ""
[2G1]   424: pi = 3.14159
[4Z3]   425: std = VAL(MID$(zeit$, 1, 2))
[9Y7]   426: IF std > 12 THEN std.alt = std - 12 ELSE st
             d.alt = std
[YT7]   427: min.alt = VAL(MID$(zeit$, 4, 2)): smin.alt
             = min.alt
[FP4]   428: sek.alt = VAL(MID$(zeit$, 7, 2))
[ØØØ]   429:
[PA1]   43Ø: Zeiger:
[8R8]   431: hilf$ = INKEY$: IF hilf$ <> "" THEN GOSUB H
             ercAus: RETURN
[VZ1]   432: zeit$ = TIME$
[4Z3]   433: std = VAL(MID$(zeit$, 1, 2))
[TA4]   434: IF std > 12 THEN std = std - 12
[BZ3]   435: min = VAL(MID$(zeit$, 4, 2))
[BZ3]   436: sek = VAL(MID$(zeit$, 7, 2))
[926]   437: IF sek <> sek.alt THEN GOSUB sekzeiger
[726]   438: IF min <> min.alt THEN GOSUB minzeiger
[846]   439: IF std <> std.alt THEN GOSUB stdzeiger
[A32]   44Ø: GOTO Zeiger
[ØØØ]   441:
[LS1]   442: sekzeiger:
[852]   443: Modepunkt% = 2
[4R1]   444: x.l1% = x.k%
[8R1]   445: y.l1% = y.k%
[6C7]   446: x.l2% = x.k% + x.r * .95 * COS(pi / 2 - pi
             * sek.alt / 3Ø)
[ZB7]   447: y.l2% = y.k% + x.r * .95 * SIN(pi / 2 - pi
             * sek.alt / 3Ø)
[4Ø2]   448: GOSUB Linie
[552]   449: Modepunkt% = 1
[4R1]   45Ø: x.l1% = x.k%
[8R1]   451: y.l1% = y.k%
[3R6]   452: x.l2% = x.k% + x.r * .95 * COS(pi / 2 - pi
             * sek / 3Ø)
[WP6]   453: y.l2% = y.k% + x.r * .95 * SIN(pi / 2 - pi
             * sek / 3Ø)
[4Ø2]   454: GOSUB Linie
[652]   455: sek.alt = sek
[552]   456: Modepunkt% = 1
[4R1]   457: x.l1% = x.k%
[8R1]   458: y.l1% = y.k%
[2U9]   459: x.l2% = x.k% + x.r * .55 * COS(pi / 2 - pi
             * (std.alt * 6Ø + smin.alt) / 36Ø)
[VT9]   46Ø: y.l2% = y.k% + x.r * .55 * SIN(pi / 2 - pi
             * (std.alt * 6Ø + smin.alt) / 36Ø)
[4Ø2]   461: GOSUB Linie
[5R2]   462: GOTO minzeiger1
[ØØØ]   463:
[ZS1]   464: minzeiger:
[852]   465: Modepunkt% = 2
[4R1]   466: x.l1% = x.k%
[8R1]   467: y.l1% = y.k%
[ZB7]   468: x.l2% = x.k% + x.r * .75 * COS(pi / 2 - pi
             * min.alt / 3Ø)
[SB7]   469: y.l2% = y.k% + x.r * .75 * SIN(pi / 2 - pi
             * min.alt / 3Ø)
[452]   47Ø: min.alt = min
[4Ø2]   471: GOSUB Linie
[ST2]   472: GOSUB stdzeiger
[CZ1]   473: minzeiger1:
[552]   474: Modepunkt% = 1
[4R1]   475: x.l1% = x.k%
[8R1]   476: y.l1% = y.k%
[WP6]   477: x.l2% = x.k% + x.r * .75 * COS(pi / 2 - pi
             * min / 3Ø)
[NP6]   478: y.l2% = y.k% + x.r * .75 * SIN(pi / 2 - pi
             * min / 3Ø)
[4Ø2]   479: GOSUB Linie
[651]   48Ø: RETURN
[ØØØ]   481:
[4T1]   482: stdzeiger:
[852]   483: Modepunkt% = 2
[4R1]   484: x.l1% = x.k%
[8R1]   485: y.l1% = y.k%
[2U9]   486: x.l2% = x.k% + x.r * .55 * COS(pi / 2 - pi
             * (std.alt * 6Ø + smin.alt) / 36Ø)
[VT9]   487: y.l2% = y.k% + x.r * .55 * SIN(pi / 2 - pi
             * (std.alt * 6Ø + smin.alt) / 36Ø)
[S52]   488: std.alt = std
[272]   489: smin.alt = min
[4Ø2]   49Ø: GOSUB Linie
[HZ1]   491: stdzeiger1:
[552]   492: Modepunkt% = 1
[4R1]   493: x.l1% = x.k%
[8R1]   494: y.l1% = y.k%
[7D8]   495: x.l2% = x.k% + x.r * .55 * COS(pi / 2 - pi
             * (std * 6Ø + min) / 36Ø)
[ØD8]   496: y.l2% = y.k% + x.r * .55 * SIN(pi / 2 - pi
             * (std * 6Ø + min) / 36Ø)
[4Ø2]   497: GOSUB Linie
[651]   498: RETURN
[ØØØ]   499:
[P62]   5ØØ: Koordinaten:
[Z83]   5Ø1: Herc.xSatz.Pos% = 35Ø
[Y83]   5Ø2: Herc.ySatz.Pos% = 25Ø
[552]   5Ø3: Modepunkt% = 1
[CØB]   5Ø4: Herc.satz$ = STR$(x.l1%) + "   " + STR$(y.l1
             %) + "   " + STR$(x.l2%) + "   " + STR$(y.l2%
             )
[NG2]   5Ø5: GOSUB HercSatz
[ØC5]   5Ø6: LPRINT sek, x.l1%, y.l1%, x.l2%, y.l2%
[651]   5Ø7: RETURN
[ØØØ]   5Ø8:
[ØØØ]   5Ø9:
[8C4]   51Ø: '---------------------------------------
[TP6]   511: 'Hier beginnen die eigentlichen Hercules-
[HP1]   512: ' Routinen
[8C4]   513: '---------------------------------------
[ØØØ]   514:
[486]   515: 'REGISTER DES VIDEO-CONTROLLERS 6845:
[VX3]   516: 'IndexReg=&HØ3B4  Data Ein
[ØY3]   517: 'ist 6845-Indexregister
[FC4]   518: 'DataReg =&HØ3B5  Data Ein/Aus
[RW3]   519: 'ist 6845-Datenregister
[AY3]   52Ø: 'ModeCont=&HØ3B8  Data Aus
[PL6]   521: 'ist 6845-Displaymode-Kontrollregister
[3Y3]   522: 'Configsw=&HØ3BF  Data Aus
[GE5]   523: 'ist 6845-Konfigurationsschalter
[ØØØ]   524:
[8C4]   525: '---------------------------------------
[P56]   526: 'Die folgende Routine initialisiert
[VG6]   527: ' die Werte für die Hercules-Routinen
[8C4]   528: '---------------------------------------
[PG1]   529: Herc.Ini:
[Z46]   53Ø: 'Grafikmodus für Seite Ø und Seite 1
[2S1]   531:  OUT &H3BF, 3
[314]   532: 'Namen zum Bild speichern
[HL3]   533:  Herc.Datei$ = "BILD.BLD"
[NB3]   534:  Herc.Seiten.mode% = 1
[PM4]   535: 'Gibt die angezeigte Seite an
[933]   536:  Herc.Video.Flag% = 1
[826]   537: 'Gibt Seite an, die beschrieben wird
[MA3]   538:  Herc.Seiten.Flag% = 1
[KW5]   539: 'Momentanes Segment zum Schreiben
[U84]   54Ø:  Herc.Seg.Schreiben% = &HB8ØØ
[WY3]   541: 'Seite 1, Videosignal ein
[5R3]   542:  Herc.Seite.Video% = &H8A
[R75]   543: 'Momentanes Segment zum Lesen
[1L3]   544:  Herc.Seg.Video% = &HB8ØØ
[8W4]   545: 'Faktor um x-Achse zu stauchen
[AD2]   546:  Herc.Apert = 1.4
[HM6]   547: 'Verzögerungswert für Delay-Routine
[ZE2]   548:  Herc.Delay = 1ØØØ
[U73]   549: '1=Segment zeichnen,
[ZN3]   55Ø: 'Ø=Kreisbogen zeichnen
[5U2]   551:  HercBogenMode% = Ø
```

Listing. Die Routinensammlung »hercules.bas« läßt sich mit Quick Basic, SBasic und Turbo Basic compilieren

```
[ØØØ]  552:
[8C4]  553: '----------------------------------------
[L86]  554: 'Alphabet in eine Zeichenkette bringen
[1R6]  555: ' Dazu sind drei Arbeitsgänge erforder-
[KH6]  556: ' lich, weil die Länge der Zeichenkette
[883]  557: ' auf 255 begrenzt ist
[8C4]  558: '----------------------------------------
[1D2]  559:   Herc.Alpha1$ = ""
[5P3]  56Ø:   RESTORE HercAlphabet1
[LP7]  561:   FOR adr1% = 1 TO (39 * 6) 'Zeichenkette ü
       bertragen
[HM1]  562:   READ alpha
[616]  563:   Herc.Alpha1$ = Herc.Alpha1$ + CHR$(alpha)
[ZSØ]  564:   NEXT
[ØØØ]  565:
[5D2]  566:   Herc.Alpha2$ = ""
[7P3]  567:   RESTORE HercAlphabet2
[LP7]  568:   FOR adr1% = 1 TO (39 * 6) 'Zeichenkette ü
       bertragen
[HM1]  569:   READ alpha
[D16]  57Ø:   Herc.Alpha2$ = Herc.Alpha2$ + CHR$(alpha)
[ZSØ]  571:   NEXT
[ØØØ]  572:
[9D2]  573:   Herc.Alpha3$ = ""
[9P3]  574:   RESTORE HercAlphabet3
[AP7]  575:   FOR adr1% = 1 TO (18 * 6) 'Zeichenkette ü
       bertragen
[HM1]  576:   READ alpha
[M16]  577:   Herc.Alpha3$ = Herc.Alpha3$ + CHR$(alpha)
[ZSØ]  578:   NEXT
[ØØØ]  579:
[A43]  58Ø: RETURN 'zum Anfang
[ØØØ]  581:
[8C4]  582: '----------------------------------------
[1Z6]  583: 'Das folgende Alphabet wird unterstützt:
[LX3]  584: ' ,!"#$%&'()*+,./Ø123456789 :;<=>?
[M55]  585: '@ABCDEFGHIJKLMNOPQRSTUVWXYZ
[MC7]  586: '[\]^_'abcdefghijklmnopqrstuvwxyz{|}~
[8C4]  587: '----------------------------------------
[SC2]  588: HercAlphabet1:
[WØ2]  589: DATA Ø,Ø,Ø,Ø,Ø,Ø
[H92]  59Ø: DATA Ø,Ø,253,Ø,Ø,Ø
[YH2]  591: DATA Ø,192,Ø,192,Ø,Ø
[SØ3]  592: DATA Ø,42,255,42,255,42
[H33]  593: DATA 36,84,215,215,84,76
[B33]  594: DATA 97,15Ø,1Ø4,54,2Ø1,6
[573]  595: DATA 1Ø2,153,153,165,66,5
[CF2]  596: DATA Ø,32,192,Ø,Ø,Ø
[SH2]  597: DATA Ø,6Ø,66,129,Ø,Ø
[2K2]  598: DATA Ø,129,66,6Ø,Ø,Ø
[TT2]  599: DATA 34,2Ø,1Ø7,2Ø,34,Ø
[U72]  6ØØ: DATA 8,8,62,8,8,Ø
[512]  6Ø1: DATA 1,6,Ø,Ø,Ø,Ø
[W22]  6Ø2: DATA Ø,8,8,8,8,Ø
[ZØ2]  6Ø3: DATA Ø,Ø,1,Ø,Ø,Ø
[7K2]  6Ø4: DATA Ø,3,12,48,192,Ø
[AA3]  6Ø5: DATA Ø,126,129,129,129,126
[DU2]  6Ø6: DATA Ø,32,64,128,255,Ø
[MD3]  6Ø7: DATA 65,131,133,137,145,97
[TL3]  6Ø8: DATA 66,129,145,145,145,11Ø
[PR2]  6Ø9: DATA 8,24,4Ø,72,255,8
[1L3]  61Ø: DATA 242,145,145,145,145,142
[AK3]  611: DATA 126,145,145,145,145,78
[6K3]  612: DATA 129,13Ø,132,136,144,224
[2K3]  613: DATA 11Ø,145,145,145,145,11Ø
[8K3]  614: DATA 114,137,137,137,137,126
[H52]  615: DATA Ø,Ø,36,Ø,Ø,Ø
[U52]  616: DATA Ø,1,38,Ø,Ø,Ø
[4G2]  617: DATA Ø,8,2Ø,34,65,Ø
[SG2]  618: DATA Ø,2Ø,2Ø,2Ø,2Ø,Ø
[EG2]  619: DATA Ø,65,34,2Ø,8,Ø
[183]  62Ø: DATA 152,164,165,164,68,Ø
[1L3]  621: DATA 127,128,158,161,161,126
[CK3]  622: DATA 127,136,136,136,136,127
[UK3]  623: DATA 255,145,145,145,145,11Ø
[FH3]  624: DATA 126,129,129,129,129,66
[MK3]  625: DATA 255,129,129,129,129,126
[BL3]  626: DATA 255,145,145,145,145,145
[6L3]  627: DATA 255,144,144,144,144,144
[UC2]  628: HercAlphabet2:
[6K3]  629: DATA 126,129,145,145,145,78
[G33]  63Ø: DATA 255,16,16,16,16,255
[ZT2]  631: DATA Ø,129,255,129,Ø,Ø
[B13]  632: DATA Ø,13Ø,129,129,254,Ø
[123]  633: DATA Ø,255,24,36,66,129
[Y92]  634: DATA 255,1,1,1,1,1
[V33]  635: DATA 255,64,32,32,64,255
[EW2]  636: DATA 255,64,48,8,6,255
[6K3]  637: DATA 126,129,129,129,129,126
[GK3]  638: DATA 255,136,136,136,136,112
[1K3]  639: DATA 126,129,129,133,131,127
[HK3]  64Ø: DATA 255,136,136,14Ø,138,113
[1F3]  641: DATA 98,145,145,145,145,78
[FA3]  642: DATA 128,128,255,128,128,Ø
[8K2]  643: DATA 254,1,1,1,1,254
[7U2]  644: DATA 24Ø,14,1,1,14,24Ø
[ØU2]  645: DATA 254,1,15,15,1,254
[Ø63]  646: DATA 129,66,52,24,1Ø2,129
[943]  647: DATA 129,66,36,24,96,128
[BK3]  648: DATA 129,131,141,145,225,129
[723]  649: DATA Ø,255,129,129,129,Ø
[MH2]  65Ø: DATA Ø,192,48,12,3,Ø
[723]  651: DATA Ø,129,129,129,255,Ø
[V23]  652: DATA 32,64,128,128,64,32
[812]  653: DATA 1,1,1,1,1,1
[PG2]  654: DATA Ø,Ø,192,32,Ø,Ø
[8V2]  655: DATA 3Ø,33,33,33,18,63
[Ø13]  656: DATA 255,33,33,33,33,3Ø
[ZU2]  657: DATA 3Ø,33,33,33,33,18
[C13]  658: DATA 3Ø,33,33,33,33,255
[NU2]  659: DATA 3Ø,41,41,41,41,18
[YT2]  66Ø: DATA Ø,8,127,136,128,Ø
[4W2]  661: DATA 25,37,37,37,37,3Ø
[UØ3]  662: DATA 255,32,32,32,32,31
[W92]  663: DATA Ø,Ø,191,Ø,Ø,Ø
[492]  664: DATA Ø,2,1,19Ø,Ø,Ø
[XN2]  665: DATA 255,4,12,18,33,Ø
[R92]  666: DATA Ø,Ø,254,1,2,Ø
[SV2]  667: DATA 16,63,32,63,32,63
[WC2]  668: HercAlphabet3:
[LP2]  669: DATA 16,63,32,32,63,Ø
[BU2]  67Ø: DATA 3Ø,33,33,33,33,3Ø
[8W2]  671: DATA 63,36,36,36,36,24
[5W2]  672: DATA 24,36,36,36,36,63
[RH2]  673: DATA Ø,63,16,32,32,Ø
[ZU2]  674: DATA 18,41,41,41,41,22
[DH2]  675: DATA Ø,32,254,33,2,Ø
[5B2]  676: DATA 62,1,1,2,62,1
[ØL2]  677: DATA 56,14,1,1,14,56
[7L2]  678: DATA 62,1,14,14,1,62
[CV2]  679: DATA 33,18,12,12,18,33
[RP2]  68Ø: DATA 33,18,12,8,16,32
[BW2]  681: DATA 33,35,37,41,49,33
[LT2]  682: DATA Ø,24,24,126,129,Ø
[L92]  683: DATA Ø,Ø,255,Ø,Ø,Ø
[NT2]  684: DATA Ø,129,126,24,24,Ø
[UH2]  685: DATA 8,16,16,8,8,16
[EM3]  686: DATA 255,255,255,255,255,255
[ØØØ]  687:
[8C4]  688: '----------------------------------------
[Ø56]  689: 'Die folgende Routine dient nur zum
[XØ6]  69Ø: ' Entwurf von Buchstaben und Zeichen
[8C4]  691: '----------------------------------------
[R72]  692: HercEntwurf:
[6D9]  693:  LINE INPUT "Ende = Leerschritt, sonst weit
       er ", Herc.satz$
[TU4]  694:  IF Herc.satz$ = " " THEN RETURN
[ØØØ]  695:
[K83]  696:  Herc.xSatz.Pos% = 1ØØ
[N83]  697:  Herc.ySatz.Pos% = 1ØØ
[XZ2]  698:  herc.Satz.Mode% = Ø
[LG2]  699:  kursiv.mode% = Ø
[L67]  7ØØ:  LINE INPUT "Buchstabe eingeben: ",Herc.Sat
       z$
[NG2]  7Ø1: GOSUB HercSatz
[B62]  7Ø2:  GOSUB HercEin
[ET1]  7Ø3: hercstabw:
[FC6]  7Ø4: a$ = INKEY$: IF a$ = "" THEN GOTO hercstabw
[H62]  7Ø5: GOSUB HercAus
[MX2]  7Ø6:  GOTO HercEntwurf
[ØØØ]  7Ø7:
[8C4]  7Ø8: '----------------------------------------
[N76]  7Ø9: 'Die folgende Routine wartet, bis eine
[MM3]  71Ø: ' Taste gedrückt wird
[8C4]  711: '----------------------------------------
[9B1]  712: Warten:
[WAØ]  713: w:
[7P5]  714:    axx$ = INKEY$: IF axx$ = "" THEN GOTO w
[651]  715: RETURN
[ØØØ]  716:
[8C4]  717: '----------------------------------------
[3Y5]  718: 'Delayx: Dient der Zeitverzögerung
[8C4]  719: '----------------------------------------
[AB1]  72Ø: Delayx:
[F18]  721:   FOR Delayx = Ø TO Herc.Delay: Delayx = De
       layx: NEXT
[651]  722: RETURN
[ØØØ]  723:
[8C4]  724: '----------------------------------------
[646]  725: 'Toggel: Schaltet zwischen den beiden
[F14]  726: ' Bildschirmen hin und her
[8C4]  727: '----------------------------------------
[EA1]  728: Toggel:
[CM3]  729: 'die Ø. Seite anwählen
[8P3]  73Ø:    Herc.Seite.Video% = &HA
[9K3]  731:    Herc.Seg.Video% = &HBØØØ
[KA3]  732:    Herc.Seiten.Flag% = Ø
[B62]  733:    GOSUB HercEin
[C32]  734:    GOSUB Delayx
[FM3]  735: 'die 1. Seite anwählen
[5R3]  736:    Herc.Seite.Video% = &H8A
[1L3]  737:    Herc.Seg.Video% = &HB8ØØ
[MA3]  738:    Herc.Seiten.Flag% = 1
[B62]  739:    GOSUB HercEin
[C32]  74Ø:    GOSUB Delayx
[651]  741: RETURN
```

```
[ØØØ]  742:
[8C4]  743: '----------------------------------------
[M36]  744: 'HercEin: Schaltet Hercules-Karte ein
[8C4]  745: '----------------------------------------
[1E1]  746: HercEin:
[8R1]  747: Grafikein:
[ØA8]  748: DATA &H35,&H2D,&H2E,&HØ7,&H5B,&HØ2,&H57,&H5
           7,&HØ2,&HØ3,&HØØ,&HØØ
[NØ3]  749:    RESTORE Grafikein
[ØØØ]  75Ø:
[Ø47]  751: 'Grafikmodus freigeben, Seite Ø/1 freigeben
[2S1]  752:    OUT &H3BF, 3
[KW5]  753: 'BIT: xØØØ 1Ø1Ø; x=Ø Seite Ø; x=1 Seite 1
[FG4]  754:    OUT &H3B8, Herc.Seite.Video%
[CX3]  755: 'BØØØ=Seite Ø, B8ØØ=Seite 1
[FV3]  756:    DEF SEG = Herc.Seg.Video%
[GX4]  757: 'Video-Controller einstellen
[723]  758:   FOR regsel% = Ø TO 11
[626]  759:    OUT &H3B4, regsel%: READ a%: OUT &H3B5,
           a%
[ZSØ]  76Ø:   NEXT
[651]  761: RETURN
[ØØØ]  762:
[8C4]  763: '----------------------------------------
[356]  764: 'HercAus: Schaltet Hercules-Karte aus
[8C4]  765: '----------------------------------------
[HF1]  766: HercAus:
[2G1]  767: Textein:
[3A8]  768:    DATA &H61,&H5Ø,&H52,&HØF,&H19,&HØ6,&H19,
           &H19,&HØ2,&HØD,&HØB,&HØC
[3X2]  769:    RESTORE Textein
[ØØØ]  77Ø:
[B77]  771: 'BIT ØØ2Ø ØØØØ: Bildschirm und Blinken frei
           geben
[Z62]  772:    OUT &H3B8, &H28
[723]  773:    FOR regsel% = Ø TO 11
[626]  774:      OUT &H3B4, regsel%: READ a%: OUT &H3B5
           , a%
[ZSØ]  775:    NEXT
[Ø91]  776:    DEF SEG
[ØØØ]  777:
[651]  778: RETURN
[ØØØ]  779:
[8C4]  78Ø: '----------------------------------------
[X36]  781: 'HercLoeschen: Löscht den Bildschirm
[8C4]  782: '----------------------------------------
[A82]  783: HercLoeschen:
[DV3]  784: 'Grafikmodus freigeben
[9N6]  785: 'Seite Ø/1 freigeben, 132 Zeichen freigeben
[2S1]  786:  OUT &H3BF, 3
[GK4]  787:  DEF SEG = Herc.Seg.Schreiben%
[4Z2]  788:    FOR adr% = Ø TO 3914
[EM2]  789:      adr2% = adr% + 3915
[RC4]  79Ø:      POKE adr%, Ø: POKE 8192 + adr%, Ø
[TC5]  791:      POKE 16384 + adr%, Ø: POKE 24576 + adr
           %, Ø
[NS4]  792:      POKE adr2%, Ø: POKE 8192 + adr2%, Ø
[NS5]  793:      POKE 16384 + adr2%, Ø: POKE 24576 + ad
           r2%, Ø
[ZSØ]  794:    NEXT
[Ø91]  795:   DEF SEG
[651]  796:  RETURN
[ØØØ]  797:
[8C4]  798: '----------------------------------------
[MW5]  799: 'HercHell: Schaltet Bildschirm hell
[8C4]  8ØØ: '----------------------------------------
[TH1]  8Ø1: HercHell:
[GK4]  8Ø2:    DEF SEG = Herc.Seg.Schreiben%
[4Z2]  8Ø3:    FOR adr% = Ø TO 3914
[EM2]  8Ø4:      adr2% = adr% + 3915
[ØØØ]  8Ø5:
[YF4]  8Ø6: 'Erste Zeile, erste Hälfte
[HA2]  8Ø7:      POKE adr%, &HFF
[FZ4]  8Ø8: 'Zweite Zeile, erste Hälfte
[413]  8Ø9:      POKE 8192 + adr%, &HFF
[LX4]  81Ø: 'Dritte Zeile, erste Hälfte
[673]  811:      POKE 16384 + adr%, &HFF
[XX4]  812: 'Vierte Zeile, erste Hälfte
[773]  813:      POKE 24576 + adr%, &HFF
[ØØØ]  814:
[CW4]  815: 'Erste Zeile, zweite Hälfte
[8C2]  816:      POKE adr2%, &HFF
[6T4]  817: 'Zweite Zeile, zweite Hälfte
[993]  818:      POKE 8192 + adr2%, &HFF
[AR4]  819: 'Dritte Zeile, zweite Hälfte
[X83]  82Ø:      POKE 16384 + adr2%, &HFF
[MR4]  821: 'Vierte Zeile, zweite Hälfte
[Y83]  822:      POKE 24576 + adr2%, &HFF
[ZSØ]  823:    NEXT
[Ø91]  824:    DEF SEG
[651]  825:  RETURN
[ØØØ]  826:
[8C4]  827: '----------------------------------------
[V86]  828: 'HercBildMode: Verknüpft Bildschirme
[ØX4]  829: '<Seite1> = <Seite1> x <SeiteØ>
[B66]  83Ø: 'x=Herc.Seiten.Mode% (Ø=AND,1=OR,2=XOR)
[8C4]  831: '----------------------------------------
[D82]  832: HercBildMode:
[4Z2]  833:    FOR adr% = Ø TO 3914
```

```
[EM2]  834:      adr2% = adr% + 3915
[ØØØ]  835:
[YF4]  836: 'Erste Zeile, erste Hälfte
[Y72]  837:      DEF SEG = &HBØØØ
[SL3]  838:      Herc.ByteØ% = PEEK(adr%)
[P82]  839:      DEF SEG = &HB8ØØ
[VL3]  84Ø:      Herc.byte1% = PEEK(adr%)
[Z9A]  841:      IF Herc.Seiten.mode% = Ø THEN POKE adr
           %, Herc.ByteØ% AND Herc.byte1%
[WBA]  842:      IF Herc.Seiten.mode% = 1 THEN POKE adr
           %, Herc.ByteØ% OR Herc.byte1%
[SCA]  843:      IF Herc.Seiten.mode% = 2 THEN POKE adr
           %, Herc.ByteØ% XOR Herc.byte1%
[ØØØ]  844:
[FZ4]  845: 'Zweite Zeile, erste Hälfte
[Y72]  846:      DEF SEG = &HBØØØ
[2F4]  847:      Herc.ByteØ% = PEEK(adr% + 8192)
[P82]  848:      DEF SEG = &HB8ØØ
[5F4]  849:      Herc.byte1% = PEEK(adr% + 8192)
[WAB]  85Ø:      IF Herc.Seiten.mode% = Ø THEN POKE adr
           % + 8192, Herc.ByteØ% AND Herc.byte1%
[R2B]  851:      IF Herc.Seiten.mode% = 1 THEN POKE adr
           % + 8192, Herc.ByteØ% OR Herc.byte1%
[7EB]  852:      IF Herc.Seiten.mode% = 2 THEN POKE adr
           % + 8192, Herc.ByteØ% XOR Herc.byte1%
[LX4]  853: 'Dritte Zeile, erste Hälfte
[Y72]  854:      DEF SEG = &HBØØØ
[6H4]  855:      Herc.ByteØ% = PEEK(adr% + 16384)
[P82]  856:      DEF SEG = &HB8ØØ
[9H4]  857:      Herc.byte1% = PEEK(adr% + 16384)
[D7B]  858:      IF Herc.Seiten.mode% = Ø THEN POKE adr
           % + 16384, Herc.ByteØ% AND Herc.byte1%
[A9B]  859:      IF Herc.Seiten.mode% = 1 THEN POKE adr
           % + 16384, Herc.ByteØ% OR Herc.byte1%
[6AB]  86Ø:      IF Herc.Seiten.mode% = 2 THEN POKE adr
           % + 16384, Herc.ByteØ% XOR Herc.byte1%
[ØØØ]  861:
[XX4]  862: 'Vierte Zeile, erste Hälfte
[Y72]  863:      DEF SEG = &HBØØØ
[HH4]  864:      Herc.ByteØ% = PEEK(adr% + 24576)
[P82]  865:      DEF SEG = &HB8ØØ
[MH4]  866:      Herc.byte1% = PEEK(adr% + 24576)
[S7B]  867:      IF Herc.Seiten.mode% = Ø THEN POKE adr
           % + 24576, Herc.ByteØ% AND Herc.byte1%
[N9B]  868:      IF Herc.Seiten.mode% = 1 THEN POKE adr
           % + 24576, Herc.ByteØ% OR Herc.byte1%
[HAB]  869:      IF Herc.Seiten.mode% = 2 THEN POKE adr
           % + 24576, Herc.ByteØ% XOR Herc.byte1%
[ØØØ]  87Ø:
[CW4]  871: 'Erste Zeile, zweite Hälfte
[Y72]  872:      DEF SEG = &HBØØØ
[UR3]  873:      Herc.ByteØ% = PEEK(adr2%)
[P82]  874:      DEF SEG = &HB8ØØ
[XR3]  875:      Herc.byte1% = PEEK(adr2%)
[PDA]  876:      IF Herc.Seiten.mode% = Ø THEN POKE adr
           2%, Herc.ByteØ% AND Herc.byte1%
[HAA]  877:      IF Herc.Seiten.mode% = 1 THEN POKE adr
           2%, Herc.ByteØ% OR Herc.byte1%
[HHA]  878:      IF Herc.Seiten.mode% = 2 THEN POKE adr
           2%, Herc.ByteØ% XOR Herc.byte1%
[ØØØ]  879:
[6T4]  88Ø: 'Zweite Zeile, zweite Hälfte
[Y72]  881:      DEF SEG = &HBØØØ
[ZG4]  882:      Herc.ByteØ% = PEEK(adr2% + 8192)
[P82]  883:      DEF SEG = &HB8ØØ
[2H4]  884:      Herc.byte1% = PEEK(adr2% + 8192)
[67B]  885:      IF Herc.Seiten.mode% = Ø THEN POKE adr
           2% + 8192, Herc.ByteØ% AND Herc.byte1%
[39B]  886:      IF Herc.Seiten.mode% = 1 THEN POKE adr
           2% + 8192, Herc.ByteØ% OR Herc.byte1%
[Z9B]  887:      IF Herc.Seiten.mode% = 2 THEN POKE adr
           2% + 8192, Herc.ByteØ% XOR Herc.byte1%
[ØØØ]  888:
[AR4]  889: 'Dritte Zeile, zweite Hälfte
[Y72]  89Ø:      DEF SEG = &HBØØØ
[5M4]  891:      Herc.ByteØ% = PEEK(adr2% + 16384)
[P82]  892:      DEF SEG = &HB8ØØ
[8M4]  893:      Herc.byte1% = PEEK(adr2% + 16384)
[1AB]  894:      IF Herc.Seiten.mode% = Ø THEN POKE adr
           2% + 16384, Herc.ByteØ% AND Herc.byte1%
[W6B]  895:      IF Herc.Seiten.mode% = 1 THEN POKE adr
           2% + 16384, Herc.ByteØ% OR Herc.byte1%
[WDB]  896:      IF Herc.Seiten.mode% = 2 THEN POKE adr
           2% + 16384, Herc.ByteØ% XOR Herc.byte1%
[ØØØ]  897:
[MR4]  898: 'Vierte Zeile, zweite Hälfte
[Y72]  899:      DEF SEG = &HBØØØ
[6M4]  9ØØ:      Herc.ByteØ% = PEEK(adr2% + 24576)
[P82]  9Ø1:      DEF SEG = &HB8ØØ
[9M4]  9Ø2:      Herc.byte1% = PEEK(adr2% + 24576)
[2AB]  9Ø3:      IF Herc.Seiten.mode% = Ø THEN POKE adr
           2% + 24576, Herc.ByteØ% AND Herc.byte1%
[X6B]  9Ø4:      IF Herc.Seiten.mode% = 1 THEN POKE adr
           2% + 24576, Herc.ByteØ% OR Herc.byte1%
[XDB]  9Ø5:      IF Herc.Seiten.mode% = 2 THEN POKE adr
           2% + 24576, Herc.ByteØ% XOR Herc.byte1%
[ZSØ]  9Ø6:    NEXT
```

Listing. Die Routinensammlung »hercules.bas« läßt sich mit Quick Basic, SBasic und Turbo Basic compilieren

```
[Ø91]  9Ø7:     DEF SEG
[651]  9Ø8:   RETURN
[ØØØ]  9Ø9:
[8C4]  91Ø: '---------------------------------------
[M36]  911: 'LoescheByte: Löscht das Byte auf der
[EC3]  912: '   Koordinate x.b%,y.b%
[RC5]  913: 'x.b%=Ø,y.b%=Ø bezeichnet die linke
[4X1]  914: ' obere Ecke
[8C4]  915: '---------------------------------------
[W42]  916: LoescheByte:
[GK4]  917:     DEF SEG = Herc.Seg.Schreiben%
[3X6]  918: 'Zunächst auf Hercules-Maße erweitern
[WX3]  919:     x.b% = x.b% * 8: y.b% = y.b% * 8
[ØR3]  92Ø:     IF x.b% > 719 THEN x.b% = 719
[ØØØ]  921:
[XK6]  922: 'In diesem Modus wird das entsprechende
[CW6]  923: ' Byte angesprochen, aber zunächst werden
[EC4]  924: ' die Grenzen überprüft
[C83]  925:     IF x.b% < Ø THEN x.b% = Ø
[YN3]  926:     IF y.b% > 346 THEN y.b% = 347
[G83]  927:     IF y.b% < Ø THEN y.b% = Ø
[ØØØ]  928:
[MX6]  929:     adr% = 8192 * (y.b% MOD 4) + 9Ø * (y.b%
       \ 4) + (x.b% \ 8)
[1R1]  93Ø:     POKE adr%, Ø
[Ø91]  931:     DEF SEG
[651]  932:   RETURN
[ØØØ]  933:
[ØØØ]  934:
[ØØØ]  935:
[8C4]  936: '---------------------------------------
[9F5]  937: 'SetzeByte: Setzt das Byte auf der
[ZC3]  938: ' Koordinate x.b%,y.b%
[RC5]  939: 'x.b%=Ø,y.b%=Ø bezeichnet die linke
[4X1]  94Ø: ' obere Ecke
[8C4]  941: '---------------------------------------
[ZU1]  942: SetzeByte:
[XV6]  943: 'Gleicher Algorithmus wie bei LoescheByte
[WX3]  944:     x.b% = x.b% * 8: y.b% = y.b% * 8
[ØR3]  945:     IF x.b% > 719 THEN x.b% = 719
[C83]  946:     IF x.b% < Ø THEN x.b% = Ø
[YN3]  947:     IF y.b% > 346 THEN y.b% = 347
[G83]  948:     IF y.b% < Ø THEN y.b% = Ø
[ØØØ]  949:
[MX6]  95Ø:     adr% = 8192 * (y.b% MOD 4) + 9Ø * (y.b%
       \ 4) + (x.b% \ 8)
[HA2]  951:     POKE adr%, &HFF
[Ø91]  952:     DEF SEG
[651]  953:   RETURN
[ØØØ]  954:
[8C4]  955: '---------------------------------------
[1C6]  956: 'ModePunkt: Bestimmt in Abhängigkeit
[YU5]  957: ' von ModePunkt%, ob Punkt gesetzt,
[CF6]  958: ' gelöscht oder mit der XOR-Funktion
[TR2]  959: ' bearbeitet wird
[8N4]  96Ø: '1=Setzen, 2=Löschen, 3=XOR
[8C4]  961: '---------------------------------------
[MU1]  962: Modepunkt:
[3T6]  963:   ON Modepunkt% GOSUB Punkt, LPunkt, XPunkt
[651]  964: RETURN
[ØØØ]  965:
[8C4]  966: '---------------------------------------
[GC6]  967: 'Check: Überprüft die Grenzen, danach
[3V5]  968: ' ist xp.p% auf Ø bis 719 und yp.p% auf
[B23]  969: ' Ø bis 347 gesetzt
[8C4]  97Ø: '---------------------------------------
[TZØ]  971: Check:
[1D5]  972: 'Dies sind die maximalen Werte der
[SU2]  973: 'Hercules-Grafik:
[MH3]  974: 'xmax%=719 und ymax%=347
[PD5]  975: 'xmin%=Ø und ymin%=Ø liegen in der
[MH6]  976: ' rechten oberen Ecke der Hercules-Karte
[Z85]  977: 'Die Parameter können bei Bedarf
[3D5]  978: ' geändert und dann per Software
[9N2]  979: ' gesetzt werden
[6D3]  98Ø:     x.null% = Ø: y.null% = Ø
[WZ5]  981: 'Lineartransformation um x.null% ...
[B43]  982:     xp.p% = x.p% + x.null%
[A85]  983: '... und um y.null% abzüglich 347,
[8D6]  984: ' so daß nun xp.p%=Ø und yp.p%=Ø in der
[PS5]  985: ' linken unteren Ecke liegen, falls
[M95]  986: ' x.null% und y.null gleich Ø ist
[ØX3]  987:     yp.p% = 347 - y.p% + y.null%
[WG6]  988: 'Es folgt das Überprüfen der Grenzen
[ZA4]  989:     IF xp.p% > 719 THEN xp.p% = 719
[PU3]  99Ø:     IF xp.p% < Ø THEN xp.p% = Ø
[RA4]  991:     IF yp.p% > 347 THEN yp.p% = 347
[VU3]  992:     IF yp.p% < Ø THEN yp.p% = Ø
[651]  993: RETURN
[ØØØ]  994:
[8C4]  995: '---------------------------------------
[WU6]  996: 'Punkt: Setzt Punkt mit Überprüfung
[K12]  997: ' der Grenzen
[8C4]  998: '---------------------------------------
[S31]  999: Punkt:
[LY1] 1ØØØ:     GOSUB Check
[VH2] 1ØØ1:     GOSUB PunktSet
[651] 1ØØ2:   RETURN
[ØØØ] 1ØØ3:
```

```
[8C4] 1Ø04: '---------------------------------------
[L25] 1Ø05: 'PunktSet: Setzt Punkt auf der
[7V3] 1Ø06: ' Koordinate xp.p%,yp.p%
[UØ6] 1Ø07: 'xp.p%=Ø,yp.p%=Ø bezeichnet die linke
[9A2] 1Ø08: ' obere Grenze
[8C4] 1Ø09: '---------------------------------------
[ZM1] 1Ø1Ø: PunktSet:
[GA5] 1Ø11: 'Die folgende Tabelle gilt global
[WØ2] 1Ø12: PktTabelle:
[G43] 1Ø13:     DATA 1,2,4,8,16,32,64,128
[4K6] 1Ø14: 'Setze Data-Zeiger aus Punkttabelle, die
[T16] 1Ø15: ' sich in der Routine HercEin befindet
[F83] 1Ø16:     RESTORE PktTabelle
[GK4] 1Ø17:     DEF SEG = Herc.Seg.Schreiben%
[SY3] 1Ø18: 'Byte-Adresse berechnen
[DS7] 1Ø19:     adr% = 8192 * (yp.p% MOD 4) + 9Ø * (yp.p
      % \ 4) + (xp.p% \ 8)
[TØ4] 1Ø2Ø: 'Bit-Position berechnen
[PW3] 1Ø21:     bitexp% = 8 - (xp.p% MOD 8)
[1Z1] 1Ø22: ' und setzen
[4Y5] 1Ø23:     FOR I% = 1 TO bitexp%: READ wert%: NEXT
[744] 1Ø24:     wert% = PEEK(adr%) OR wert%
[XF2] 1Ø25:     POKE adr%, wert%
[Ø91] 1Ø26:     DEF SEG
[651] 1Ø27: RETURN
[ØØØ] 1Ø28:
[8C4] 1Ø29: '---------------------------------------
[2H7] 1Ø3Ø: 'XPunkt: Verknüpft Punkt mit XOR-Funktion
[365] 1Ø31: ' bei Überprüfung der Grenzen
[8C4] 1Ø32: '---------------------------------------
[WD1] 1Ø33: XPunkt:
[LY1] 1Ø34:     GOSUB Check
[9W2] 1Ø35:     GOSUB XPunktSet
[651] 1Ø36:   RETURN
[ØØØ] 1Ø37:
[8C4] 1Ø38: '---------------------------------------
[TL6] 1Ø39: 'XPunktSet: Verknüpft Punkt auf der
[BU6] 1Ø4Ø: ' Koordinate xp.p%,yp.p% mit XOR-Funktion
[UØ6] 1Ø41: 'xp.p%=Ø,yp.p%=Ø bezeichnet die linke
[4X1] 1Ø42: ' obere Ecke
[8C4] 1Ø43: '---------------------------------------
[VX1] 1Ø44: XPunktSet:
[WC6] 1Ø45: 'Gleicher Algoritmus wie bei PunktSet
[F83] 1Ø46:     RESTORE PktTabelle
[GK4] 1Ø47:     DEF SEG = Herc.Seg.Schreiben%
[DS7] 1Ø48:     adr% = 8192 * (yp.p% MOD 4) + 9Ø * (yp.p
      % \ 4) + (xp.p% \ 8)
[PW3] 1Ø49:     bitexp% = 8 - (xp.p% MOD 8)
[4Y5] 1Ø5Ø:     FOR I% = 1 TO bitexp%: READ wert%: NEXT
[R84] 1Ø51:     wert% = PEEK(adr%) XOR wert%
[XF2] 1Ø52:     POKE adr%, wert%
[Ø91] 1Ø53:     DEF SEG
[651] 1Ø54: RETURN
[ØØØ] 1Ø55:
[8C4] 1Ø56: '---------------------------------------
[V47] 1Ø57: 'LPunkt: Löscht Punkt auf der Koordinate
[DØ7] 1Ø58: ' x.p%,y.p% mit Überprüfung der Grenzen
[P26] 1Ø59: 'x.p%=Ø,y.p%=Ø bezeichnet linke untere
[AVØ] 1Ø6Ø: ' Ecke
[8C4] 1Ø61: '---------------------------------------
[4D1] 1Ø62: LPunkt:
[LY1] 1Ø63:   GOSUB Check
[R22] 1Ø64:   GOSUB PRset
[651] 1Ø65:   RETURN
[ØØØ] 1Ø66:
[8C4] 1Ø67: '---------------------------------------
[SZ6] 1Ø68: 'PRset: Löscht Punkt auf der Koordinate
[YU1] 1Ø69: ' xp.p%,yp.p%
[UØ6] 1Ø7Ø: 'xp.p%=Ø,yp.p%=Ø bezeichnet die linke
[4X1] 1Ø71: ' obere Ecke
[8C4] 1Ø72: '---------------------------------------
[X31] 1Ø73: PRset:
[WC6] 1Ø74: 'Gleicher Algoritmus wie bei PunktSet
[F83] 1Ø75:     RESTORE PktTabelle
[GK4] 1Ø76:     DEF SEG = Herc.Seg.Schreiben%
[DS7] 1Ø77:     adr% = 8192 * (yp.p% MOD 4) + 9Ø * (yp.p
      % \ 4) + (xp.p% \ 8)
[PW3] 1Ø78:     bitexp% = 8 - (xp.p% MOD 8)
[4Y5] 1Ø79:     FOR I% = 1 TO bitexp%: READ wert%: NEXT
[Z25] 1Ø8Ø:     wert% = PEEK(adr%) AND (255 - wert%)
[XF2] 1Ø81:     POKE adr%, wert%
[Ø91] 1Ø82:     DEF SEG
[651] 1Ø83:   RETURN
[ØØØ] 1Ø84:
[8C4] 1Ø85: '---------------------------------------
[BL6] 1Ø86: 'Es folgen die Routinen für die Grund-
[786] 1Ø87: ' figuren Kreis, Kreisbogen und Linie
[8C4] 1Ø88: '---------------------------------------
[ØØØ] 1Ø89:
[8C4] 1Ø9Ø: '---------------------------------------
[1F6] 1Ø91: 'Kreis: Zeichnet oder löscht Kreis mit
[2X5] 1Ø92: ' dem Mittelpunkt auf der Koordinate
[RH4] 1Ø93: ' x.k%,y.k% und dem Radius x.r
[8C4] 1Ø94: '---------------------------------------
[X11] 1Ø95: Kreis:
[KM3] 1Ø96:   b = Ø!: bmax = x.r * .7Ø71Ø68
[ØØØ] 1Ø97:
[4K1] 1Ø98: Kreisanf:
[SM3] 1Ø99:     z = SQR(x.r * x.r - b * b)
```

```
[ØØØ] 11ØØ:
[WD7] 11Ø1:       x1 = x.k% + z: x2 = x.k% + b: x3 = x.k%
                - z: x4 = x.k% - b
[ØD7] 11Ø2:       y1 = y.k% + b: y2 = y.k% + z: y3 = y.k%
                - b: y4 = y.k% - z
[ØØØ] 11Ø3:
[SR7] 11Ø4:       x.p% = x1 * Herc.Apert: y.p% = y1: GOSU
              B Modepunkt
[WR7] 11Ø5:       x.p% = x2 * Herc.Apert: y.p% = y2: GOSU
              B Modepunkt
[ØS7] 11Ø6:       x.p% = x3 * Herc.Apert: y.p% = y3: GOSU
              B Modepunkt
[4S7] 11Ø7:       x.p% = x4 * Herc.Apert: y.p% = y4: GOSU
              B Modepunkt
[WR7] 11Ø8:       x.p% = x3 * Herc.Apert: y.p% = y1: GOSU
              B Modepunkt
[ØS7] 11Ø9:       x.p% = x4 * Herc.Apert: y.p% = y2: GOSU
              B Modepunkt
[WR7] 111Ø:       x.p% = x1 * Herc.Apert: y.p% = y3: GOSU
              B Modepunkt
[ØS7] 1111:       x.p% = x2 * Herc.Apert: y.p% = y4: GOSU
              B Modepunkt
[ØØØ] 1112:
[8Ø6] 1113:       b = b + 1: IF b < bmax THEN GOTO Kreisa
              nf
[ØØØ] 1114:
[651] 1115:     RETURN
[ØØØ] 1116:
[8C4] 1117: '---------------------------------------
[PT6] 1118: 'Bogen: Zeichnet oder löscht Kreisbogen
[7H6] 1119: ' mit dem Mittelpunkt auf der Koordinate
[K36] 112Ø: ' x.k%,y.k% und dem Radius x.r von α bis
[8P5] 1121: ' ß in Grad (d.h. Ø bis 36Ø) und dem
[YF3] 1122: ' Winkelinkrement dwi
[8C4] 1123: '---------------------------------------
[WØ1] 1124: Bogen:
[YS7] 1125:       x.anf% = CINT(x.k + x.Achse * COS(alph
              a * 3.14159 / 18Ø))
[PU7] 1126:       y.anf% = CINT(y.k + y.Achse * SIN(alph
              a * 3.14159 / 18Ø))
[XG7] 1127:       x.end% = CINT(x.k + x.Achse * COS(beta
               * 3.14159 / 18Ø))
[NK7] 1128:       y.end% = CINT(y.k + y.Achse * SIN(beta
               * 3.14159 / 18Ø))
[PG2] 1129:       x.mp% = CINT(x.k)
[VG2] 113Ø:       y.mp% = CINT(y.k)
[HP6] 1131:       FOR herc.winkel = alpha TO beta STEP d
              wi
[XS5] 1132:       herc.hilf = herc.winkel * 3.14159 / 18
              Ø
[4F8] 1133:       x.p% = CINT((x.k + x.Achse * COS(herc.
              hilf)) * Herc.Apert)
[9N6] 1134:       y.p% = CINT(y.k + y.Achse * SIN(herc.h
              ilf))
[WT2] 1135:       GOSUB Modepunkt
[ZSØ] 1136:     NEXT
[TZ5] 1137:     IF HercBogenMode% = Ø THEN GOTO Bogen3
[5PA] 1138:     x.11% = x.mp%: y.11% = y.mp%: x.12% = x.a
              nf%: y.12% = y.anf%: GOSUB Linie
[5PA] 1139:     x.11% = x.mp%: y.11% = y.mp%: x.12% = x.e
              nd%: y.12% = y.end%: GOSUB Linie
[F71] 114Ø: Bogen3:
[651] 1141:   RETURN
[ØØØ] 1142:
[8C4] 1143: '---------------------------------------
[ØB6] 1144: 'Linie: Zeichnet oder löscht Linie von
[914] 1145: ' x.11%,y.11% bis x.12%,y.12%
[8C4] 1146: '---------------------------------------
[C11] 1147: Linie:
[GT5] 1148: 'Damit ein Quadrat auch wirklich ein
[GF6] 1149: ' Quadrat wird, wird mit Aperturfaktor
[LT2] 115Ø: ' multipliziert
[SR4] 1151:     x.11% = CINT(x.11% * Herc.Apert)
[VR4] 1152:     x.12% = CINT(x.12% * Herc.Apert)
[6R4] 1153: 'Vertikale Linie, dann Sprung
[M45] 1154:     IF x.11% <> x.12% THEN GOTO Linie2
[1FC] 1155:     IF y.11% = y.12% THEN x.p% = x.11%: y.p%
              = y.11%: GOSUB Modepunkt: RETURN 'nur Punkt
[EC3] 1156: 'nur vertikale Linie
[RM1] 1157:     x.p% = x.11%
[MNB] 1158:     IF y.11% < y.12% THEN FOR y.p% = y.11% TO
               y.12%: GOSUB Modepunkt: NEXT: RETURN
[U59] 1159:     FOR y.p% = y.12% TO y.11%: GOSUB Modepunk
              t: NEXT: RETURN
[ØØØ] 116Ø:
[XF6] 1161: 'Es sind nur horizontale und schräge
[613] 1162: ' Linien zulässig
[W71] 1163: Linie2:
[2C5] 1164:   IF y.11% <> y.12% THEN GOTO Linie3:
[5F2] 1165: 'herc.hilf=y.11%
[TH2] 1166: 'y.p% = herc.hilf
[DRD] 1167:   IF x.11% < x.12% THEN FOR x.p% = x.11% TO
              x.12%: y.p% = y.11%: GOSUB Modepunkt: NEXT:
               RETURN
[LUA] 1168:   FOR x.p% = x.12% TO x.11%: y.p% = y.11%: G
              OSUB Modepunkt: NEXT: RETURN
[ØØØ] 1169:
[WK6] 117Ø: 'Es sind nur schräge Linien zulässig
[Z71] 1171: Linie3:
```

```
[6DB] 1172:     IF (y.11% > y.12% AND x.11% > x.12%) THEN
               GOTO Linie4 'Komponenten tauschen
[5Z1] 1173:     GOTO Linie5
[281] 1174: Linie4:
[7G7] 1175:     herc.hilf% = y.12%: y.12% = y.11%: y.11%
              = herc.hilf%
[ZF7] 1176:     herc.hilf% = x.12%: x.12% = x.11%: x.11%
              = herc.hilf%
[ØØØ] 1177:
[581] 1178: Linie5:
[PZ7] 1179:     m = (y.12% - y.11%) / (x.12% - x.11%) 'St
              eigung der Linien
[ØØØ] 118Ø:
[CC6] 1181:       IF m > Ø THEN GOTO Linie6 'Linien ste
              igend
[DZ1] 1182:       GOTO Linie7
[ØØØ] 1183:
[VD3] 1184: Linie6: 'unter 45 Grad
[3T3] 1185:     IF m < 1 THEN GOTO Linie61
[VØ2] 1186:     GOTO Linie62
[Y81] 1187: Linie61:
[P95] 1188:       herc.hilf = y.11%: y.p% = herc.hilf
[PR3] 1189:       FOR x.p% = x.11% TO x.12%
[WT2] 119Ø:     GOSUB Modepunkt
[FH6] 1191:     herc.hilf = herc.hilf + m: y.p% = herc.
              hilf
[ZSØ] 1192:       NEXT
[651] 1193:       RETURN
[ØØØ] 1194:
[PF5] 1195: Linie62: 'über oder gleich 45 Grad
[UF6] 1196:       herc.hilf = x.11%: x.p% = herc.hilf: m
               = 1 / m
[VR3] 1197:       FOR y.p% = y.11% TO y.12%
[8E9] 1198:     GOSUB Modepunkt: herc.hilf = herc.hilf
              + m: x.p% = herc.hilf
[ZSØ] 1199:       NEXT
[651] 12ØØ:       RETURN
[ØØØ] 12Ø1:
[RY3] 12Ø2: Linie7: 'Linien fallend
[GX9] 12Ø3:   IF m < -1! THEN GOTO Linie71 'Linien stärk
              er als 45 Grad fallend
[ZØ2] 12Ø4:   GOTO Linie72
[191] 12Ø5: Linie71:
[SUB] 12Ø6:       IF y.11% > y.12% THEN GOTO Linie711 'E
              s wird von unten nach oben gezeichnet
[242] 12Ø7:       GOTO Linie712
[CC1] 12Ø8: Linie711:
[4D7] 12Ø9:       herc.hilf = y.11%: y.11% = y.12%: y.12
              % = herc.hilf
[346] 121Ø:       herc.hilf = x.12%: m = 1 / m: GOTO Lin
              ie713
[DC1] 1211: Linie712:
[GT3] 1212:       herc.hilf = x.11%: m = 1 / m
[ØØØ] 1213:
[EC1] 1214: Linie713:
[TF2] 1215:       x.p% = herc.hilf
[VR3] 1216:       FOR y.p% = y.11% TO y.12%
[8E9] 1217:     GOSUB Modepunkt: herc.hilf = herc.hilf
              + m: x.p% = herc.hilf
[X52] 1218:       NEXT: RETURN
[ØØØ] 1219:
[591] 122Ø: Linie72:
[91C] 1221:       IF x.11% > x.12% THEN GOTO Linie721 'E
              s wird von links nach rechts gezeichnet
[342] 1222:       GOTO Linie722
[GC1] 1223: Linie721:
[WC7] 1224:       herc.hilf = x.11%: x.11% = x.12%: x.12
              % = herc.hilf
[ZZ4] 1225:       herc.hilf = y.12%: GOTO Linie723
[HC1] 1226: Linie722:
[NL2] 1227:       herc.hilf = y.11%
[KC1] 1228: Linie723:
[VF2] 1229:       y.p% = herc.hilf
[PR3] 123Ø:       FOR x.p% = x.11% TO x.12%
[AE9] 1231:     GOSUB Modepunkt: herc.hilf = herc.hilf
              + m: y.p% = herc.hilf
[ZSØ] 1232:       NEXT
[651] 1233:       RETURN
[ØØØ] 1234:
[8C4] 1235: '---------------------------------------
[956] 1236: 'HercSpeichern: Speichert oder lädt
[9V2] 1237: ' Bildschirmseite
[8C4] 1238: '---------------------------------------
[BF2] 1239: HercSpeichern:
[PØ2] 124Ø: herc.hilf% = Ø
[L78] 1241: INPUT "Grafik speichern = 1, sonst laden",
              herc.hilf%
[VK5] 1242: PRINT "Dateiname ist "; Herc.Datei$
[Y37] 1243: INPUT "Dateinamen ändern (j oder n): ", hil
              f$
[MKA] 1244: IF hilf$ = "j" OR hilf$ = "J" THEN INPUT "N
              euer Dateiname: ", Herc.Datei$
[181] 1245: hilf% = 1
[AX5] 1246: INPUT "Seite 1 = 1 sonst Seite Ø ", hilf%
[WZ5] 1247: IF herc.hilf% = 1 THEN GOTO Speichern3
[WSA] 1248: IF hilf% = 1 THEN DEF SEG = &HB8ØØ: BLOAD H
              erc.Datei$, Ø: GOTO Speichern2
[759] 1249: DEF SEG = &HBØØØ: BLOAD Herc.Datei$, Ø: BEE
              P: GOTO Speichern2
```

Listing. Die Routinensammlung »hercules.bas« läßt sich mit Quick Basic, SBasic und Turbo Basic compilieren

```
[MY1] 125Ø: Speichern3:
[T6B] 1251: IF hilf% = Ø THEN DEF SEG = &HØ: BSAVE Herc
            .Datei$, Ø, 32768: GOTO Speichern2
[6D6] 1252: DEF SEG = &HB8ØØ: BSAVE Herc.Datei$, Ø, 327
            68
[ØØØ] 1253:
[HY1] 1254: Speichern2:
[Ø91] 1255: DEF SEG
[651] 1256: RETURN
[ØØØ] 1257:
[8C4] 1258: '---------------------------------------
[M76] 1259: 'HercSatz: Gibt Satz aus, dazu müssen
[XF5] 126Ø: ' folgende Parameter bekannt sein:
[VK6] 1261: 'Herc.Satz$ = Zeichenkette, die den Satz
[WN1] 1262: ' enthält
[Y26] 1263: 'Herc.xSatz.Pos% = Satzbeginn auf der
[3C1] 1264: ' x-Achse
[236] 1265: 'Herc.ySatz.Pos% = Satzbeginn auf der
[7C1] 1266: ' y-Achse
[GY5] 1267: 'Herc.Satz.Mode% = Ø: Satz horizontal
[FD6] 1268: 'Herc.Satz.Mode% = 1: Satz vertikal von
[AN4] 1269: ' oben nach unten geschrieben
[8C4] 127Ø: '---------------------------------------
[HL1] 1271: HercSatz:
[GM4] 1272:   IF Herc.satz$ = "" THEN RETURN
[2P6] 1273:   herc.xpos% = Herc.xSatz.Pos% * Herc.Apert
[9S4] 1274:   herc.ypos% = Herc.ySatz.Pos%
[XL4] 1275:   satzlaenge% = LEN(Herc.satz$)
[CG4] 1276:   FOR hilfs% = 1 TO satzlaenge%
[926] 1277:     buchstabe$ = MID$(Herc.satz$, hilfs%, 1
            )
[RN2] 1278:     GOSUB buchstabe
[LØC] 1279:    IF herc.Satz.Mode% = 1 THEN herc.ypos% =
             herc.ypos% - 1Ø: GOTO HercSatzWeiter
[E64] 128Ø:     herc.xpos% = herc.xpos% + 8
[UU2] 1281: HercSatzWeiter:
[ZSØ] 1282:   NEXT
[651] 1283: RETURN
[ØØØ] 1284:
[8C4] 1285: '---------------------------------------
[MR6] 1286: 'Buchstabe: Schreibt auf die Koordinate
[HS5] 1287: ' x,y einen Buchstaben in einer 8x6-
[X91] 1288: ' Matrix
[MA6] 1289: 'Der Aufruf erfolgt mit buchstabe$="A"
[L75] 129Ø: ' oder einem anderen Buchstaben
[NB6] 1291: 'herc.xpos% enthält die x-Position im
[SE3] 1292: ' Bereich von Ø bis 711
[ØU6] 1293: 'herc.ypos% enthält die y-Position im
[9F3] 1294: ' Bereich von Ø bis 345
[G85] 1295: 'kursivmode% = Ø: Normalschrift
[7A5] 1296: 'kursivmode% = 1: Kursivschrift
[LY5] 1297: '(Bei der Initialisierung wurden die
[ØY5] 1298: ' Daten der Buchstaben in die String-
[CG4] 1299: ' Variablen Herc.Alpha1$ bis
[1X3] 13ØØ: ' Herc.Alpha3$ übertragen
[8C4] 13Ø1: '---------------------------------------
[RR1] 13Ø2: buchstabe:
[WS7] 13Ø3: 'Zwischenspeicher für Daten eines Buchstabe
            ns
[ØX1] 13Ø4:   DIM alpha(6)
[AY1] 13Ø5:   xkursiv = Ø
[GU5] 13Ø6:   IF kursiv.mode% = 1 THEN xkursiv = .25
[7N3] 13Ø7:   hilf% = ASC(buchstabe$)
[257] 13Ø8: 'Für Buchstaben, die nicht vorhanden sind
[H26] 13Ø9:   IF hilf% < 32 OR hilf% > 135 THEN hilf% =
             128
[VNB] 131Ø:   IF hilf% > 1Ø9 THEN Herc.Alpha$ = Herc.Al
            pha3$: hilf% = hilf% - 11Ø: GOTO Buchw
[P5B] 1311:   IF hilf% > 7Ø THEN Herc.Alpha$ = Herc.Alp
            ha2$: hilf% = hilf% - 71: GOTO Buchw
[AR6] 1312:   Herc.Alpha$ = Herc.Alpha1$: hilf% = hilf%
             - 32
[ØØØ] 1313:
[L11] 1314: Buchw:
[FT6] 1315: 'Bestimmen der Position des Buchstabens
[453] 1316:   stelle% = hilf% * 6 + 1
[ØØØ] 1317:
[TW6] 1318: 'Übertragen der Daten in einen Speicher
[6X2] 1319:     FOR hilf% = Ø TO 5:
[RH6] 132Ø:       hilf$ = MID$(Herc.Alpha$, stelle% + hi
            lf%, 1)
[BM3] 1321:     alpha(hilf%) = ASC(hilf$)
[ZSØ] 1322:     NEXT
[PH3] 1323: 'Buchstaben ausgeben
[4N2] 1324:   FOR adr1% = Ø TO 5
[EP3] 1325:   x.p% = herc.xpos% + adr1%
[6X1] 1326:   xreal = x.p%
[PN3] 1327:   herc.hilf% = alpha(adr1%)
[BN2] 1328:     FOR adr2% = Ø TO 7
[LP3] 1329:       y.p% = herc.ypos% + adr2%
[PY3] 133Ø:       xreal = xreal + xkursiv
[HT1] 1331:       x.p% = xreal
[U63] 1332:       hilf% = herc.hilf% \ 2
[455] 1333:       Modepunkt% = herc.hilf% - 2 * hilf%
[N3C] 1334:       IF Modepunkt% = 1 THEN GOSUB Modepunk
            t: GOTO Buchstabeweiter 'Bildpunkt
[864] 1335: 'Bildpunkt wird gelöscht
[C75] 1336:       Modepunkt% = 2: GOSUB Modepunkt
[4X2] 1337: Buchstabeweiter:
[9P2] 1338:       herc.hilf% = hilf%
[ZSØ] 1339:     NEXT
[ZSØ] 134Ø:   NEXT
[651] 1341:  RETURN
[ØØØ] 1342:
[8C4] 1343: '---------------------------------------
[956] 1344: 'HercApertur: Dehnt oder staucht die
[2R5] 1345: ' x-Achse, um das Seitenverhältnis
[G14] 1346: ' einstellen zu können
[8C4] 1347: '---------------------------------------
[B72] 1348: HercApertur:
[5Ø8] 1349: PRINT "Aperturfaktor ist momentan: "; Herc.
            Apert
[Z6B] 135Ø: INPUT "Aperturfaktor :", Herc.Apert 'Faktor
            , um x-Achse zu stauchen
[651] 1351: RETURN
[ØØØ] 1352:
[8C4] 1353: '---------------------------------------
[E46] 1354: 'HercDrucken: Gibt Hardcopy der Bild-
[2N4] 1355: ' schirmseite auf Drucker aus
[8C4] 1356: '---------------------------------------
[P62] 1357: HercDrucken:
[8Y2] 1358: GOSUB HercSpiegel
[12B] 1359: REM WIDTH "lpt1:", 255 'Bei Turbo-Basic und
             Quick Basic REM entfernen
[B18] 136Ø:     WIDTH                'Diese Zeile nur bei
             SBasic ohne REM
[33A] 1361:     PRINT "Druckausgabe läuft, mit Tastendr
            uck zu unterbrechen"
[FV3] 1362:    DEF SEG = Herc.Seg.Video%
[2P7] 1363:    LPRINT CHR$(27); "3"; CHR$(8); 'Zeilenvo
            rschub 24/18Ø"
[EA6] 1364:    LPRINT                          ' einmal
            ausführen
[BF7] 1365:     FOR a1% = Ø TO 89               'Es gibt
            9Ø Byte horizontal
[BFA] 1366:     LPRINT CHR$(27); "*"; CHR$(33); CHR$(92
            ); CHR$(1); '92+256=348 Druckspalten
[ZØ8] 1367:      FOR a2% = Ø TO 347             '348 Zeil
            en werden übertragen
[UR5] 1368:  adr% = (a2% MOD 4) * 8192 + (a2% \ 4) * 9Ø
             + a1%
[AY5] 1369:  LPRINT CHR$(PEEK(adr%)); CHR$(Ø); CHR$(Ø);
[ZSØ] 137Ø:      NEXT
[K17] 1371:     'Jetzt ist eine Byte-Spalte mit 348 Zei
            len
[U35] 1372:     'untereinander zu übertragen
[H51] 1373:     LPRINT
[6A7] 1374:     a$ = "": a$ = INKEY$: IF a$ <> "" THEN
            GOTO Druckaus
[ZSØ] 1375:     NEXT
[AC5] 1376:     'Es sind alle Spalten bearbeitet
[9M1] 1377: Druckaus:
[ØX2] 1378:    LPRINT CHR$(27); "2";
[8Y2] 1379:    GOSUB HercSpiegel
[Ø91] 138Ø:    DEF SEG
[651] 1381:  RETURN
[ØØØ] 1382:
[8C4] 1383: '---------------------------------------
[CH5] 1384: 'HercSpiegel: Spiegelt das Bild an
[VM6] 1385: ' der Mittelsenkrechten; dies ist nötig
[WH3] 1386: ' für die Druckausgabe
[8C4] 1387: '---------------------------------------
[G52] 1388: HercSpiegel:
[FV3] 1389:    DEF SEG = Herc.Seg.Video%
[B62] 139Ø:    GOSUB HercEin
[TD2] 1391:    FOR a1% = Ø TO 347
[HSA] 1392:    zeile% = (a1% MOD 4) * 8192 + (a1% \ 4)
            * 9Ø 'Byteadresse des 1. Byte der Zeile
[S82] 1393:      FOR a2% = Ø TO 44
[VU3] 1394:       hilf1% = PEEK(zeile% + a2%)
[SA4] 1395:       hilf2% = PEEK(zeile% + 89 - a2%)
[L22] 1396:       hilf% = hilf2%
[9G2] 1397:       GOSUB Reversal
[TF3] 1398:       POKE zeile% + a2%, hilf%
[H22] 1399:       hilf% = hilf1%
[9G2] 14ØØ:       GOSUB Reversal
[9Z3] 14Ø1:       POKE zeile% + 89 - a2%, hilf%
[ZSØ] 14Ø2:      NEXT
[ZSØ] 14Ø3:    NEXT
[H62] 14Ø4:    GOSUB HercAus
[Ø91] 14Ø5:    DEF SEG
[651] 14Ø6: RETURN
[ØØØ] 14Ø7:
[8C4] 14Ø8: '---------------------------------------
[Ø36] 14Ø9: 'BitReversal: Invertiert das Byte in
[HU5] 141Ø: ' hilf%, Ergebnis wird wieder in hilf%
[XV2] 1411: ' zurückgegeben
[8C4] 1412: '---------------------------------------
[HL1] 1413: Reversal:
[WZ1] 1414:  rev.zahl% = 1
[BR2] 1415:  rev.hilf1% = hilf%
[Z71] 1416:  hilf% = Ø
[EV1] 1417: RevWeiter:
[134] 1418:  rev.hilf2% = rev.hilf1% \ 2
[MK6] 1419:  hilf% = hilf% * 2 + rev.hilf1% - 2 * rev.h
            ilf2%
[2L3] 142Ø:  rev.hilf1% = rev.hilf2%
[EU3] 1421:  rev.zahl% = rev.zahl% + 1
[ØP5] 1422:  IF rev.zahl% < 9 THEN GOTO RevWeiter
[651] 1423:  RETURN
```

Listing. Die Routinensammlung »hercules.bas« läßt sich mit Quick Basic, SBasic und Turbo Basic compilieren

Noch erhältlich:

DOS EXTRA Nr. 3

Datenfernübertragungsprogramm in 1000% Assemblercode
Hercules-Grafik-Toolbox in Turbo Pascal
Grafikfunktionen für Turbo C
Zeichensatzeditor für Epson-kompatible Drucker
Treiber für AT-Tastatur
Alle Listings zum Kurs »PC-glasklar«
Demo für Sprachdigitalisierung

DOS EXTRA Nr. 4

Desktop-Publishing-Programm »SDTP«
Kommandoeditor für MS-DOS
Editor für Disketten und Festplatten
Speicherresidente Programme auflisten und entfernen
Paßwort-Schutz für MS-DOS
Tastatureingabe automatisiert
Clock-Utility für alle Grafikkarten
Cursor und Zeichensatz für EGA-Karte
Fenstertechnik für Turbo C
Generator für Hilfsbildschirme

DOS EXTRA Nr. 5

166 Farben für die CGA-Farbgrafikkarte
Snapshot: Bildschirmfotos per Software
Druckerzeichensatz selbstgemacht
Mausgesteuertes Zeichenprogramm
Editor in GW-Basic
Setup-Programm für ATs
Maschinensprache für GW-Basic
Erklärtes Listing: das Basic-Spiel »Stardos«

DOS EXTRA Nr. 6

SBasic-Compiler im Quelltext
Datenbankprogramm »Parat«
Fensterorientierte Funktionen für GW-Basic und SBasic
Turbo-Basic-Bibliothek für 16 Farben
Analyse von Grafikkarten
Maskengenerator für Binärdateien
dBase-Datenbanken in Basic
Superhirn für Quick Basic 4.0

DOS EXTRA Nr. 7

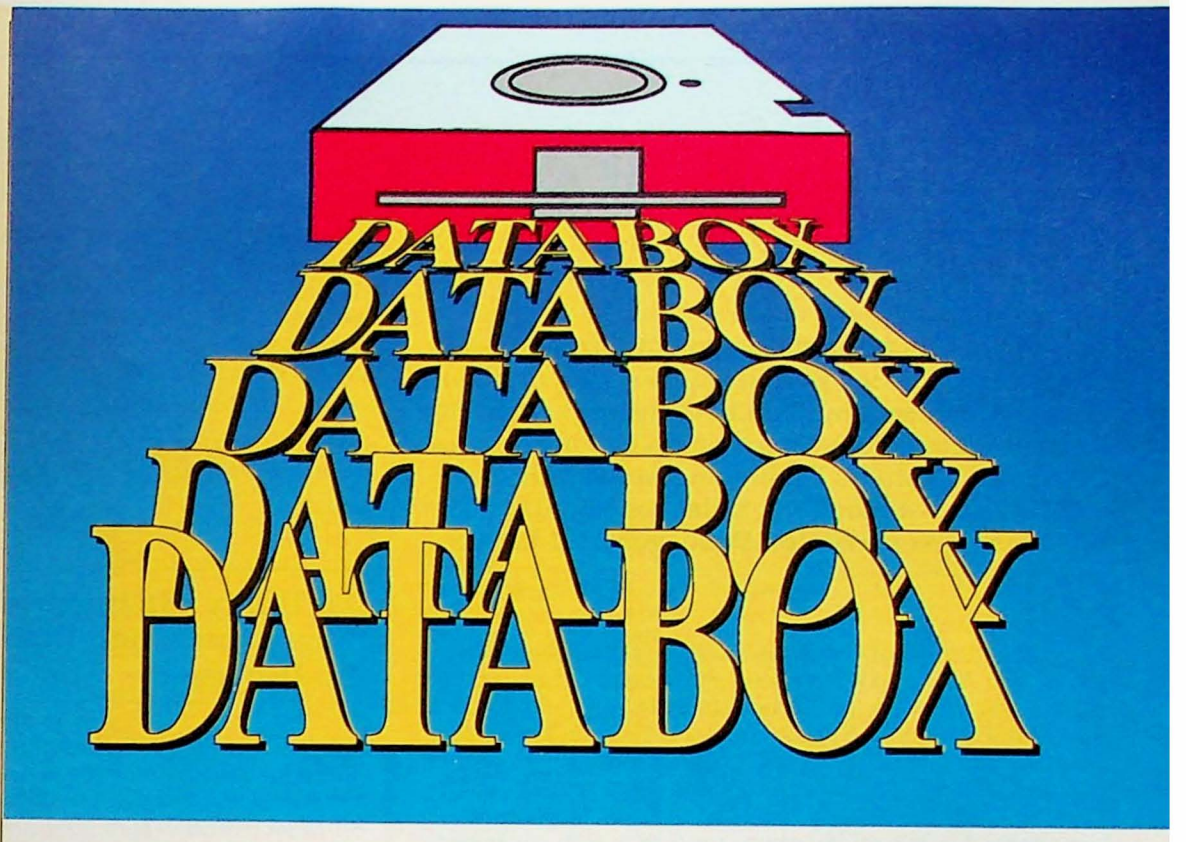
MS-DOS-Benutzeroberfläche Easydos
Benutzeroberfläche und Präprozessor für SBasic
Datengenerator für Basic, C und Pascal
Sprite-Editor für Farbgrafikkarten
DOS-Packer
Virenschutzprogramm
Weltkarte im Computer
Schneller Zugriff auf der CGA-Karte
Kurven einfacher plotten
Einfacher Prozessor-Emulator
sowie alle Listings zu den Grundlagenbeiträgen und den Tips & Tricks

DOS EXTRA Nr. 8

Makroassembler SAM III mit vielen Beispielen
Exe2bin-Ersatz durch "exe2com"
Programm zur Verbesserung der Speicherverwaltung unter MS-DOS
Routine zum Speichern von Hardcopies
Alle Listings zum Einsteigerschwerpunkt und zu den Werkstätten

DOS EXTRA Nr. 9

Definitive Eingabe für Batchdateien
Minidatenbank per Batchdatei
Utility für komfortable MS-DOS-Befehlseingabe
Utility für sicheres Dateilöschen
Pulldown-Menüs unter GW-Basic programmieren
Briefbogen selbstgemacht
Analoguhr für Selfmade-Designer
Konvertierung vom EGA- in den Hercules-Modus: Schraffur statt Farbe
sowie die Listings aller anderen Beiträge



Die Software zu den DOS-EXTRA-Heften

Sagenhaft günstig, extra viel gute Software fürs Geld!

DATABOX:

- mehr als der übliche Softwareservice
- bringt alle Programme der DOS EXTRA auf Datenträger

Für alle PCs unter MS-DOS

Den Aufbau der Databox-Diskette finden Sie in der Datei »read.me« erklärt. Geben Sie deshalb vor dem Start »type read.me« ein. Sämtliche Pascal- und C-Programme liegen auch in einer compilierten Form für Hercules- und Farbgrafikkarten vor. Es ist allerdings nicht gewährleistet, daß diese Versionen auf Hardware, die nicht vollständig kompatibel zum IBM PC ist, problemlos laufen. In diesem Fall müssen Sie den Quellcode nach der Anpassung mit dem entsprechenden Compiler neu übersetzen.
Die Basic-Programme arbeiten mit dem GW-Basic-Interpreter von Microsoft einwandfrei. Bei Interpretern und Compilern anderer Hersteller kann es zu Problemen bei der Programmausführung kommen, sofern diese nicht voll kompatibel sind.

jeweils 2 Disketten 5 1/4"
1 Diskette 3 1/2"

35,– DM (unverbindliche Preisempfehlung)
Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	35,– DM	Einzelpreis	35,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**39,– DM**	**Endpreis**	**41,– DM**

Unabhängig von der Anzahl der bestellten Produkte berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung.

– Bitte benutzen Sie die Bestellkarte. –

DOS EXTRA Nr. 10

- Units-Bibliotheken für Turbo Pascal: Datum, Sound, Bildschirmausgabe
- Prüflisting für die Leistungsdaten der Hardware
- Snapshot-Utility, Version 2.7
- TSR-Datei-Manager
- DOS-Assembler SAM III, Version 1.1
- Schnelles Kopierprogramm mit 115200 Bit pro Sekunde
- Komfortable Dateiumleitung
- Hercules-Grafik in Basic programmiert
- Patch für die BGI-Treiber von Turbo Pascal
- Verbesserungen zu SBASIC

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

SBasic unter der Lupe

Haben Sie schon einmal versucht, für den Basic-Compiler »SBasic« Erweiterungen zu schreiben? Hatten Sie dann nicht auch Schwierigkeiten, Ihre Routinen nach dem Compilieren zu »debuggen«? Diese und ähnliche Probleme löst das Utility »SB2map« fast spielend, indem es Map-Dateien für den weitverbreiteten Debugger AFD-Pro erzeugt.

Einer der komfortabelsten und leistungsfähigsten Debugger ist der AFD-Pro (Advanced Fullscreen Debugger) von Adtec. Der AFD-Pro (Bild) besitzt zum Beispiel einen Befehl zum Laden von Symbolen aus einer vom Linker erzeugten Map-Datei. Weil beim SBasic-Compiler kein Linken möglich ist, mußte man bislang ohne solch eine, das Debuggen sehr vereinfachende Map-Datei auskommen.
Um dieses Manko zu beheben, muß man sich zunächst einmal überlegen, welche Daten sinnvollerweise durch eine Map-Datei zur Verfügung gestellt werden sollten. Hier sind zunächst die Routinen zu nennen, die SBasic aus der Laufzeitbibliothek »sbasic.bin« in ein compiliertes Programm kopiert. Als Symbolname könnte man einfach die Marke (Label) nehmen, ab der die entsprechende Routine beginnt (zum Beispiel: »L01«).
Weil diese Marken aber wenig aussagekräftig sind, ist es besser, eine eigene Liste von Namen, die meist mit dem entsprechenden Befehl übereinstimmen, zu benutzen. So entspricht die Marke »L01« im Listing von »sbasic.asm« (siehe DOS Extra Ausgabe 6) beispielsweise dem Symbol »End«, das schon wesentlich mehr aussagt.
Außerdem ist es wichtig, die Adressen zu kennen, ab der eine bestimmte Basic-Programmzeile beginnt. Weil der AFD-Pro aber ein Symbol, das mit einer Ziffer beginnt, nicht erkennt, müssen diese Zeilenangaben in Zeichenketten umgesetzt werden, die mit dem Buchstaben »Z« beginnen. So können Sie zum Beispiel ein Programm an der compilierten Zeile 100 mit dem AFD-Pro-Befehl »g,z100« unterbrechen.

Map-Dateien vereinfachen das Debuggen sehr – leider bei SBasic bislang unmöglich

Sinnvoll ist es ferner, die Adressen von Variablen und benutzerdefinierten Funktionen (»def fn«) zu kennen.
Weil ein Utility, das Map-Dateien erzeugen soll, aufgrund der genannten Anforderungen Informationen über die Startadressen der einzelnen Routinen und Basic-Zeilen benötigt und diese nur vom SBasic-Compiler beim Compilieren berechnet werden, ist eine Erweiterung von SBasic unumgänglich. Hierzu tippen Sie unter GW-Basic Listing 1 ab und speichern es mit

```
save "sbneu.asc",a
```

als ASCII-Datei.
Die erweiterte Version von SBasic erhalten Sie mit Hilfe des Merge-Befehls, der zwei einzelne Programmdateien zu einer großen Datei zusammenbindet:

```
load "sbasic
merge "sbneu.asc
save "sbasic
```

Die neu erzeugte SBasic-Version müssen Sie nach Verlassen von GW-Basic nur noch mit

```
sbasic sbasic
```

vom »alten« SBasic compilieren lassen.
Wenn Sie alles richtig gemacht haben, dann versteht SBasic jetzt den Parameter »/m«. Diese Option, die bereits beim Aufruf von SBasic in der Kommandozeile angegeben werden muß, darf sowohl vor als auch hinter einem Dateinamen stehen. Die Syntax des SBasic-Aufrufs darf also

```
sbasic [/m] [Dateiname]
```

und

```
sbasic [Dateiname] [/m]
```

lauten.

Das Utility »SB2map« erzeugt aus SBasic-Programmen Map-Dateien für den AFD-Pro-Debugger

Bei der Angabe des Parameters »/m« erzeugt SBasic eine Datei mit der Erweiterung »sb«, die die für den AFD-Pro benötigten Informationen enthält. Die Angaben in einer SB-Datei befinden sich aber noch nicht in der für den AFD-Pro erforderlichen Form. Für die Konvertierung einer SB-Datei in eine für den AFD-Pro geeignete Map-Datei ist daher das SBasic-Utility »SB2map« zuständig.
Tippen Sie dazu das Programm (Listing 2) ab, und speichern Sie es diesmal im Token-Format:

```
save "sb2map
system
```

Um einen ersten Eindruck von den Fähigkeiten von SB2map zu erhalten, geben Sie folgende Befehle ein:

```
sbasic sb2map /m
sb2map sb2map
```

Mit dieser Befehlsfolge erzeugen Sie die beiden Dateien »sb2map.sb« und »sb2map.map« sowie gleichzeitig das ausführbare SB2map-Utility »sb2map.exe«. Die Datei »sb2map.sb« ist dabei nur eine Zwischendatei und darf nach Aufruf von SB2map getrost wieder gelöscht werden.
Anschließend starten Sie den AFD-Pro und laden das Programm SB2map. Nach Eingabe von »ls« lädt der Debugger automatisch die Datei »sb2map.map«. Sie sehen nun zum Beispiel, daß der Befehl »Call 0000« durch »Call Init« ersetzt wurde. »Init« ist die Initialisierungsroutine, die im Assembler-Listing von SBasic ab der Marke »L00« be-

Der AFD-Pro ist ein äußerst komfortabler und leistungsfähiger Fullscreen-Debugger

Name	Marke	Bemerkung
Asc	L37	
Atn	L98	
Beep	L83	
Bload	L86	
Bsave	L85	
Call	L3D	
ChDir	L81	
Chr$	L19	
Clear	L7E	
Close	L20	
CloseAll	L26	CLOSE ohne Dateinummern
Cls	L17	
Color	L1C	
Command$	L94	
Cos	L96	
Csrlin	L5C	
Cvi	L6D	
Cvs	L6E	
Date$	L6B	
Date$Befehl	L93	DATE$ = ...
End	L01	auch SYSTEM
Eof	L27	
Exp	L91	
Field	L40	
Files	L8D	
Fix	L6F	
Fn$	L34	Rückkehr von Fn$
FnInt	L33	Rückkehr von Fn%
FnSng	L58	Rückkehr von Fn!
Get	L41	
GetBackColr	L63	ermittle Hintergrundfarbe (COLOR)
GetCurOnOff	L5F	ermittle, ob Cursor sichtbar (LOCATE)
GetCurZeileOb	L60	lies erste Cursor-Zeile (LOCATE)
GetCurZeileUn	L61	lies letzte Cursor-Zeile (LOCATE)
GetForeColr	L62	ermittle Vordergrundfarbe (COLOR)
Hex$	L70	
Init	L00	Initialisierung
Inkey$	L28	
Inp	L71	
Input	L24	
Input$	L2F	
Instr	L31	
Int	L50	
Int2Sng	L45	
Int2SngZwOb	L49	für Funktionen, entspricht in etwa Int2Sng
Int2Str	L16	STR$
IntAbs	L8A	Integer-Funktion ABS
IntAdd	L08	Integer-Funktion +
IntAnd	L0A	Integer-Funktion AND
IntDiv	L0E	Integer-Funktion /
IntEq	L04	Integer-Vergleich =
IntGe	L05	Integer-Vergleich > =
IntGt	L02	Integer-Vergleich >
IntLe	L06	Integer-Vergleich < =
IntLt	L03	Integer-Vergleich <
IntMod	L0F	Integer-Funktion MOD
IntMul	L0D	Integer-Funktion *
IntNe	L07	Integer-Vergleich < >
IntNeg	L11	Integer-Vorzeichenwechsel
IntNot	L10	Integer-Funktion NOT
IntOr	L0B	Integer-Funktion OR
IntSgn	L1A	Integer-Funktion SGN
IntSub	L09	Integer-Funktion -
IntSwap	L77	Integer-Befehl SWAP
IntXor	L0C	Integer-Funktion XOR
Kill	L8C	
LeerStr2Stack	L7B	Push LeerString
Left$	L38	
Len	L25	
LineInput	L22	
Locate	L1B	
Lof	L99	
Log	L4C	
Lset	L42	
Mid$	L3A	
Mid$Befehl	L3E	MID$ =

Name	Marke	Bemerkung
MkDir	L7F	
Mki$	L67	
Mks$	L68	
Name	L8E	
NeueZeile	L30	gehe nach Befehl INPUT in neue Zeile
OnGosub	L4F	
OnGoto	L4E	
Open	L1F	
Out	L88	
Peek	L36	
Play	L89	
Poke	L35	
Pos	L5D	
Print	L15	
Print14Tab	L21	PRINT Komma
PrintCrLf	L14	PRINT ohne abschließendes Semikolon
PrintFragezeichen	L23	für INPUT
Put	L44	
Randomize	L82	
Right$	L39	
RmDir	L80	
Rnd	L69	
Rset	L43	
Screen	L6A	Funktion SCREEN
ScreenBefehl	L7D	
SetCurForm	L5E	für LOCATE
SetRandFarbe	L87	für COLOR
Shell	L7C	
Sin	L95	
Sng2Int	L51	
Sng2IntZwOb	L65	
Sng2Str	L46	STR$
SngAbs	L8B	(Vergleiche ab Marke »IntAbs«)
SngAdd	L48	
SngDiv	L4B	
SngEq	L54	
SngGe	L55	
SngGt	L52	
SngLe	L56	
SngLt	L53	
SngMul	L4A	
SngNe	L57	
SngNeg	L47	
SngPot	L4D	A^B
SngSgn	L59	
SngSwap	L78	
Sound	L7A	
Space$	L84	
Spc	L5B	
Sqr	L92	
Stick	L8F	
Str2Speicher	L13	
Str2Stack	L12	
StrAdd	L18	String-Funktionen und Vergleiche beginnen mit »Str«
StrEq	L29	
StrGe	L2D	
StrGt	L2B	
Strig	L90	
StrInFn	L32	
String$	L3B	
StrLe	L2C	
StrLt	L2A	
StrNe	L2E	
StrSwap	L79	
Tab	L5A	
Tan	L97	
TestLeerStr	L66	
Time$	L3F	
Time$Befehl	L76	TIME$ = ...
Timer	L6C	
Val	L64	
VarPtrStr	L3C	VARPTR(A$)
ViewPrint	L1E	
Wait	L75	
Width	L1D	
WriteKomma	L74	
WriteStr	L73	
WriteZahl	L72	

Die Tabelle zeigt die von »SB2map« benutzten Symbolmarken

ginnt. Die Symbole »Z100« bis »Z2036« entsprechen dem Beginn der Programmzeilen 100 bis 2036.
Im einzelnen stellt eine Map-Datei folgende Informationen zur Verfügung:

- Die Adressen der aufgerufenen Bibliotheksfunktionen und deren lesbare Namen, zum Beispiel »Input«. Die entsprechenden Marken »Lnn« entnehmen Sie der Tabelle.
- Die Adressen der Zeilen im compilierten Programm. Die Zeilennummern sind dem Debugger als »Zn« bekannt, wobei »n« für die ursprüngliche Zeilennummer steht.
- Die Adressen der allgemeinen Variablen und deren Bezeichnungen (entspricht den ersten 48 Byte im DS-Segment)
- Die Adressen der übrigen Variablen: Integer-Variablen in der Form »Word__Int«, Single-Variablen in der Form »Pl__Sng«, String-Variablen in der Form »Text$__Str« und Feldvariablen in den Formen »Liste__IntFeld«, »Tabelle__SngFeld« sowie »Zeilen$__StrFeld«.
- Die Adressen der benutzerdefinierten Funktionen, zum Beispiel »Fnword__Int«, »Fnreal__Sng« und »Fntxt$__Str«

Wenn Ihnen die Bezeichnungen der Bibliotheksfunktionen nicht gefallen, können Sie diese selbstverständlich ändern. Die Namen finden Sie in Listing 2 in den Data-Zeilen ab Programmzeile 1000. Aber auch dem Hinzufügen von Namen eigener Routinen steht nichts im Wege. Schreiben Sie diese einfach in eine neue Data-Zeile. Entsprechendes gilt für die Bezeichnungen und Längen der allgemeinen Variablen, die in den Data-Zeilen ab Programmzeile 2000 stehen. Dabei enthält jede Zeile zunächst den Namen der Variablen und dahinter, durch ein Komma getrennt, die Länge dieser Variablen. Sie müssen nur darauf achten, daß die Gesamtlänge aller Variablen 48 Byte betragen muß.

Das Debuggen von SBasic-Programmen wird mit »SB2map« und dem AFD-Pro fast zum Kinderspiel

Wenn Sie SB2map mit größeren Programmen testen, kann es vorkommen, daß der Debugger einmal den Fehler »Symboltabelle voll« anzeigt. Der Debugger lädt dann nur einen Teil der Symbole der Map-Datei. Diese Meldung bekommen Sie leider schon dann, wenn Sie versuchen, den SBasic-Compiler selbst mit sämtlichen Symbolen zu laden.

(Torsten Harling/ma)

Name	sbneu.asc
Sprache	GW-Basic, SBasic
Grafikkarte	jede
Besonderheiten	Erweiterung für SBasic, die SB-Dateien für das SBasic-Utility »SB2map« erzeugt

```
[FBC] 1Ø3 AUS$=SPACE$(255):EINDAT$=COMMAND$(1):IF EINDA
      T$="/M" THEN EINDAT$=COMMAND$(2)
[XS3] 1Ø4 IF EINDAT$<>"" THEN 1Ø7
[SC7] 1Ø5 PRINT:INPUT "Quelldatei [.BAS]: ",EINDAT$
[MZ4] 185 REM *** SB-Datei schreiben ***
[8Ø8] 186 IF COMMAND$(1)<>"/M" AND COMMAND$(2)<>"/M" TH
      EN END
[FT5] 187 PRINT:PRINT "Schreibe SB-Datei"
[SC7] 188 AUSDAT$=LEFT$(EINDAT$,INSTR(EINDAT$,"."))+"SB
      "
[BB5] 189 OPEN AUSDAT$ FOR OUTPUT AS #1
[NM4] 19Ø WRITE #1,RC!,PC!,VC!,UPMAX
[YY6] 191 FOR I=Ø TO UPMAX-1:WRITE #1,UPADR(I):NEXT
[2R2] 192 WRITE #1,ZEILE
[FX8] 193 FOR I=Ø TO ZEILE-1:WRITE #1,ZEILNR(I),ZEILADR
      (I):NEXT
[353] 194 WRITE #1,VARIABLE
[VG9] 195 FOR I=Ø TO VARIABLE-1:WRITE #1,VARNAME$(I),VA
      RADR(I):NEXT
[6T2] 196 WRITE #1,FUNNR
[P49] 197 FOR I=Ø TO FUNNR-1:WRITE #1,FUNNAME$(I),FUNAN
      F(I):NEXT
[MN1] 198 CLOSE 1
[N41] 199 END
```

Listing 1. Die SBasic-Erweiterung »sbneu.asc« erlaubt das Anlegen einer SB-Datei mit Informationen für das Utility »SB2map«

Name	sb2map.bas
Sprache	GW-Basic, SBasic
Grafikkarte	jede
Besonderheiten	Utility für SBasic zum Erzeugen von Map-Dateien für den Debugger AFD-Pro

```
[KPG] 1ØØ REM SB2map 1.Ø, (C) 1989 Redaktionsbüro Evert
      s&Hagedorn, DOS International, Autor: Torsten Har
      ling
[61C] 1Ø1 PRINT "SB2map 1.Ø, Utility zum Erzeugen von M
      ap-Dateien aus SB-Dateien"
[B2C] 1Ø2 PRINT "     (C) 1989 Redaktionsbüro Everts&Ha
      gedorn, DOS International"
[KT8] 1Ø3 DEF FNH$(X)=RIGHT$("ØØØ"+HEX$(X),4): EINDAT$=
      COMMAND$(1)
[1T3] 1Ø4 IF EINDAT$<>"" THEN 1Ø8
[PC1] 1Ø5 PRINT
[N75] 1Ø6 INPUT "SB-Datei [.SB]";EINDAT$
[BR3] 1Ø7 IF EINDAT$="" THEN END
[6K4] 1Ø8 REM *** Dateien öffnen ***
[S38] 1Ø9 IF INSTR(EINDAT$,".")=Ø THEN EINDAT$=EINDAT$+
      ".SB"
[DL7] 11Ø AUSDAT$=LEFT$(EINDAT$,INSTR(EINDAT$,"."))+"MA
      P"
[PV4] 111 OPEN EINDAT$ FOR INPUT AS #1
[5BA] 112 IF EOF(1) THEN PRINT "Datei ";EINDAT$;" nicht
       gefunden!":BEEP:END
[ØA5] 113 OPEN AUSDAT$ FOR OUTPUT AS #2
[EC8] 114 REM *** Benutzte Bibliotheksfunktionen lesen
      ***
[2L3] 115 INPUT #1,RC,PC,VC,UP
[5G2] 116 FOR I=1 TO UP
[4A2] 117 INPUT #1,ADR
[3K1] 118 READ A$
[UL2] 119 IF I=1 THEN 121
[PW2] 12Ø IF ADR=Ø THEN 122
[VF6] 121 PRINT #2," ";FNH$(Ø);":"FNH$(ADR);SPC(7);A$
[Z71] 122 NEXT
[SE5] 123 REM *** Zeilennummern lesen ***
[BB2] 124 INPUT #1,CNT
[FY2] 125 FOR I=1 TO CNT-1
[L13] 126 INPUT #1,ZNR,ADR
[ZV9] 127 PRINT #2," ";FNH$(Ø);":";FNH$(RC+ADR);SPC(7);
      "Z";MID$(STR$(ZNR),2)
[R81] 128 NEXT
[CU3] 129 REM *** End Of Code ***
[YØ3] 13Ø INPUT #1,ZNR,ADR
[2K8] 131 PRINT #2," ";FNH$(Ø);":";FNH$(RC+ADR);SPC(7);
      "EndOfCode"
[KA6] 132 REM *** Allgemeine Variablen lesen ***
[BL4] 133 VARSEG=INT((RC+PC+15)/16)+256
[WW4] 134 STRSEG=VARSEG+INT((VC+15)/16)
[HZ2] 135 I=Ø:RESTORE 2ØØØ
[322] 136 WHILE I<48
[992] 137 READ NAM$,L
[2F7] 138 PRINT #2," ";FNH$(VARSEG);":";FNH$(I);SPC(7);
      NAM$
[F91] 139 I=I+L
[R61] 14Ø WEND
[SN4] 141 REM *** Variablen lesen ***
[9B2] 142 INPUT #1,CNT
[8R2] 143 FOR I=1 TO CNT
[FØ3] 144 INPUT #1,NAM$,ADR
[LV2] 145 A$=RIGHT$(NAM$,1)
[TG4] 146 IF A$<>"(" THEN D$="":GOTO 15Ø
[RS1] 147 D$="Feld"
[SP4] 148 NAM$=LEFT$(NAM$,LEN(NAM$)-1)
[3W2] 149 A$=RIGHT$(NAM$,1)
[VS2] 15Ø IF A$="$" THEN 155
[ZN4] 151 NAM$=LEFT$(NAM$,LEN(NAM$)-1)
```

```
[M78] 152 IF A$="!" THEN NAM$=NAM$+"_Sng" ELSE NAM$=NAM
      $+"_Int"
[358] 153 PRINT #2," ";FNH$(VARSEG);":";FNH$(ADR);SPC(7
      );NAM$;D$
[8R1] 154 GOTO 156
[VN9] 155 PRINT #2," ";FNH$(STRSEG+ADR);":";FNH$(Ø);SPC
      (7);NAM$;"_Str";D$
[S81] 156 NEXT
[M78] 157 REM *** Benutzerdefinierte Funktionen lesen *
      **
[4C2] 158 INPUT #1,CNT
[3S2] 159 FOR I=1 TO CNT
[5Ø3] 160 INPUT #1,NAM$,ADR
[9V2] 161 A$=RIGHT$(NAM$,1)
[C86] 162 IF A$="$" THEN NAM$=NAM$+"_Str":GOTO 165
[AP4] 163 NAM$=LEFT$(NAM$,LEN(NAM$)-1)
[Z78] 164 IF A$="!" THEN NAM$=NAM$+"_Sng" ELSE NAM$=NAM
      $+"_Int"
[Z18] 165 PRINT #2," ";FNH$(Ø);":";FNH$(RC!+ADR);SPC(7)
      ;"Fn";NAM$
[V81] 166 NEXT
[3C1] 167 CLOSE
[841] 168 END
[FF7] 1ØØØ REM *** Daten für Bibliotheksfunktionen ***
[CR6] 1ØØ1 DATA Init,End,IntGt,IntLt,IntEq,IntGe
[EY6] 1ØØ2 DATA IntLe,IntNe,IntAdd,IntSub,IntAnd
[MZ5] 1ØØ3 DATA IntOr,IntXor,IntMul,IntDiv
[8L6] 1ØØ4 DATA IntMod,IntNot,IntNeg,Str2Stack
[AA6] 1ØØ5 DATA Str2Speicher,PrintCrLf,Print
[ZA6] 1ØØ6 DATA Int2Str,Cls,StrAdd,Chr$,IntSgn
[GA6] 1ØØ7 DATA Locate,Color,Width,ViewPrint
[AR6] 1ØØ8 DATA Open,Close,Print14Tab,LineInput
[S46] 1ØØ9 DATA PrintFragezeichen,Input,Len
[1L6] 1Ø1Ø DATA CloseAll,Eof,Inkey$,StrEq,StrLt
[AT6] 1Ø11 DATA StrGt,StrLe,StrGe,StrNe,Input$
[UN6] 1Ø12 DATA NeueZeile,Instr,StrInFn,FnInt
[366] 1Ø13 DATA Fn$,Poke,Peek,Asc,Left$,Right$
[8S5] 1Ø14 DATA Mid$,String$,VarPtrStr,Call
[6A6] 1Ø15 DATA Mid$Befehl,Time$,Field,Get,Lset
[2E6] 1Ø16 DATA Rset,Put,Int2Sng,Sng2Str,SngNeg
[D47] 1Ø17 DATA SngAdd,Int2SngZwOb,SngMul,SngDiv
[BP6] 1Ø18 DATA Log,SngPot,OnGoto,OnGosub,Int
[KS6] 1Ø19 DATA Sng2Int,SngGt,SngLt,SngEq,SngGe
[HR6] 1Ø2Ø DATA SngLe,SngNe,FnSng,SngSgn,Tab,Spc
[3E7] 1Ø21 DATA Csrlin,Pos,SetCurForm,GetCurOnOff
[Ø86] 1Ø22 DATA GetCurZeileOb,GetCurZeileUn
[NZ5] 1Ø23 DATA GetForeColr,GetBackColr,Val
[U27] 1Ø24 DATA Sng2IntZwOb,TestLeerStr,Mki$,Mks$
[Y86] 1Ø25 DATA Rnd,Screen,Date$,Timer,Cvi,Cvs
[SL6] 1Ø26 DATA Fix,Hex$,Inp,WriteZahl,WriteStr
[6Y5] 1Ø27 DATA WriteKomma,Wait,Time$Befehl
[BT6] 1Ø28 DATA IntSwap,SngSwap,StrSwap,Sound
[DY6] 1Ø29 DATA LeerStr2Stack,Shell,ScreenBefehl
[267] 1Ø3Ø DATA Clear,MkDir,RmDir,ChDir,Randomize
[BØ5] 1Ø31 DATA Beep,Space$,Bsave,Bload
[M66] 1Ø32 DATA SetRandFarbe,Out,Play,IntAbs
[B56] 1Ø33 DATA SngAbs,Kill,Files,Name,Stick
[4W5] 1Ø34 DATA Strig,Exp,Sqr,Date$Befehl
[3P5] 1Ø35 DATA Command$,Sin,Cos,Tan,Atn,Lof
[SG1] 1999 DATA *
[D67] 2ØØØ REM *** Daten für allgemeine Variablen ***
[TA3] 2ØØ1 DATA HandleScrn,1
[GF3] 2ØØ2 DATA HandleFile1,1
[LF3] 2ØØ3 DATA HandleFile2,1
[PF3] 2ØØ4 DATA HandleFile3,1
[963] 2ØØ5 DATA HandlePrn,1
[5X2] 2ØØ6 DATA PosFile1,1
[9X2] 2ØØ7 DATA PosFile2,1
[DX2] 2ØØ8 DATA PosFile3,1
[TP2] 2ØØ9 DATA PosPrn,1
[ET3] 2Ø1Ø DATA ScrnAttribut,1
[2E2] 2Ø11 DATA RFlag,1
[UC3] 2Ø12 DATA RandRechts,1
[UP3] 2Ø13 DATA ViewPrintOb,1
[TR3] 2Ø14 DATA ViewPrintUn,1
[Z93] 2Ø15 DATA FilePuffer,1
[ZT3] 2Ø16 DATA CurFormStart,2
[EH3] 2Ø17 DATA NxtChrFile1,1
[FH3] 2Ø18 DATA NxtChrFile1,1
[KH3] 2Ø19 DATA NxtChrFile2,1
[FH3] 2Ø2Ø DATA NxtChrFile3,1
[XS3] 2Ø21 DATA CurFormInput,2
[SA3] 2Ø22 DATA DataZeiger,2
[T62] 2Ø23 DATA DSeg,2
[7W2] 2Ø24 DATA LenFile1,1
[BW2] 2Ø25 DATA LenFile2,1
[FW2] 2Ø26 DATA LenFile3,1
[TH3] 2Ø27 DATA FeldInfFile1,2
[XH3] 2Ø28 DATA FeldInfFile2,2
[1K3] 2Ø29 DATA FeldInfFile3,2
[6B3] 2Ø3Ø DATA CurFormAkt,2
[WX2] 2Ø31 DATA LastRnd,4
[DN2] 2Ø32 DATA PspSeg,2
[LN2] 2Ø33 DATA Oktave,1
[YK3] 2Ø34 DATA NotenLaenge,1
[GD3] 2Ø35 DATA MusikTempo,1
[R53] 2Ø36 DATA TonLaenge,1
```

Listing 2. Das Utility »SB2map.bas« erzeugt Map-Dateien für den Debugger AFD-Pro

Der Umwandler

Spezial-Adapter zum Anschluß aller Drucker mit COMMODORE-IEC-Schnittstelle (seriell) an die PARALLELE Drucker-schnittstelle des PC

(wie beispielsweise der LC-10, NL-10, MPS-801, SP 180 VC, SP 1200 VC und viele mehr)

Commodore-Drucker jetzt auch am PC

PRN-IEC-Adapter
DM 69,-
zuzügl. Versandk.

Lieferumfang: Adapterkabel, Installations-Software (5,25" oder 3,5" bei Bestellung angeben) und Bedienungsanleitung

Alleinvertrieb:
Rubel EDV-Applikationen
Oberweg 14a,
6368 Bad Vilbel 3,
Telefon: 06101/4 34 60
(oder 069/42 43 21)

Der Umwandler Der Umwandler Der Umwandler

Nürnberger Public Domain Studio
Humboldtstr.141 im CCC
8500 Nürnberg 40
Tel.:0911/45 77 54 Fax : 0911 446 72 79

Ca.9000 Public Domain + Shareware Disketten für Amiga , Atari ST und PC.
Gratis Katalog - Disk und Info für Ihren Computer anfordern !!!
Sig - 1690, Blue - 585, German PD - 520, Deutsche - 580, DB_PP - 51, Turbo, TU, DB, General, CUG, USA, Utility, Clipper Tools, Unicorn, Quick Basic, Turbo Basic Tools, Modula 2, Peepshow.

Farbige Disketten

5.25" 2DD

	10 Stück	100 Stück
Schwarz	DM5.90	DM54.90
Weiss	DM9.90	DM94.90
Rot	DM9.90	DM94.90
Grau	DM9.90	DM94.90
Gelb	DM9.90	DM94.90
Blau	DM9.90	DM94.90
Orange	DM9.90	DM94.90

3.5" 2DD

	10 Stück	50 Stück
Blau	DM11.90	DM55.00
Grau	DM11.90	DM55.00
Rot	DM17.90	DM85.00
Gelb	DM17.90	DM85.90
Grün	DM17.90	DM85.90

5.25" HD
10 Stück 5.25" HD DM 12.80

LEERDISKETTEN SIND NUR ÜBER VERSAND ZU BEZIEHEN
Versandkosten:
Nachnahme DM 6.- Vorkasse DM3.-

Softfair proudly presents

GABI'S magisches Kaleidoskop

Fantastische Farben und Formen

RS ROULETTE

Super Heim – Spiel – Bank
UNTERHALTUNG
für 1 – 9 Spieler
SYSTEMENTWICKLUNG
mit Datenbank, Prognose Statistik und Schnellauf
SIMULATION
mit vorgegebenen Zahlen "Was wäre wenn ..."

■ KALEIDOSKOP *verträumte* 50 DM
■ ROULETTE *spielerische* 100 DM

für MS – DOS ab Ver. 2.1 mit Hercules – , EGA – , oder VGA – Grafik – Karte

beide Programme 135 DM
Versandkosten 6 DM
per Nachnahme / Scheck
Diskettenformat angeben !

Raimund Scholz
Bindingstraße 10
6000 Frankfurt 70

PC – Software zu fairen Preisen

Verbesserungen zu SBasic

Mittlerweile ist der SBasic-Compiler aus der DOS Extra 6 seit einem Jahr im praktischen Einsatz. Parallel dazu haben die DOS-Leser aber auch einige Fehlfunktionen entdeckt.

Die erste Version eines Compilers hat immer ihre Tücken. Bei den folgenden Korrekturen für die SBasic-Laufzeitbibliothek »sbasic.asm« wird davon ausgegangen, daß Sie die in DOS 10'89 vorgestellten Verbesserungen bereits ausgeführt haben. Andernfalls müssen Sie dies nachholen, weil sonst die hier angegebenen Zeilennummern des Assembler-Listings zum Teil nicht stimmen.
Ein Fehler von SBasic ist in der Ausgaberoutine von Zeile 333 bis 340 zu finden. Hier wird die Cursor-Bewegung nach links behandelt. In Zeile 335 steht ein Sprung auf Zeile 353, der ausgeführt wird, wenn der Cursor bereits am linken Rand steht. Weil aber ab Zeile 353 das Zeichen im AL-Register ausgegeben wird und das BIOS dieses Zeichen nicht unterstützt, liegt hier ein Fehler vor. Korrekt wäre ein Sprung auf Zeile 339. Deshalb müssen Sie die Zeile 335 und 339 ändern. Im folgenden nun die korrigierten Zeilen ab Programmzeile 333:

```
;Am linken Rand?
CMP DL,0
JE XOON1
DEC DL
JMP XOOI
;Eine Zeile nach oben gehen
XOON1: MOV DL,DS:[11]
       JMP SHORT XOOM
```

Bei der Subtraktion großer Zahlen tritt in SBasic ein Arithmetikfehler auf. Dieser läßt sich durch einfaches Ändern der Programmzeile 884 beheben. Sie lauten dann:

```
AND DX,DX
```

Des weiteren gibt es Fehler in der Routine für den Files-Befehl. Wenn Sie beispielsweise in einem compilierten Programm den Befehl

```
FILES "A:*.*"
```

benutzen und das aktuelle Laufwerk die Festplatte C ist, dann sind die angezeigten Dateien zwar von Laufwerk A, die Angabe der freien Bytes und des Pfades stammen jedoch vom aktuellen Laufwerk C.
Um diesen Fehler zu beheben, ändern Sie die Programmzeilen 5611 bis 5613 (vor der Marke »X8DK«) wie folgt:

```
JNE X8DJ
XOR BX,BX
NEG CX
```

Zusätzlich müssen Sie den Teil der Files-Routine von Zeile 5394 (vor der Marke »L8D«) bis 5472 (vor dem Kommentar »;In neue Zeile gehen«) durch die Programmzeilen aus Listing 1 ersetzen. Für die Berechnung der Quadratwurzel (»sqr«) wurde eine neue Routine entwickelt, die die Wurzel nicht nur schneller, sondern auch genauer als die Originalroutine berechnet. Listing 2 zeigt die neue Routine, die im Assemblerquelltext die Programmzeilen 5817 bis 5850 ersetzt.
Für die Leser, die keinen Assembler besitzen und den Hexdump der SBasic-Laufzeitbibliothek aus DOS Extra 6 abgetippt haben, zeigt Listing 3 den Hexdump der neuen Laufzeitbibliothek.
Auch für den Basic-Quelltext »sbasic.bas« gibt es eine Verbesserung. Wird beispielsweise die Befehlsfolge

```
OPEN F$ AS 1 LEN=128
```

compiliert, erhalten Sie eine Fehlermeldung, weil SBasic zwar das Nummernzeichen »#« vor der Dateinummer nicht zwingend verlangt, GW-Basic aber die Angabe »1« in diesem Fall anders codiert. Abhilfe schafft eine Änderung der Programmzeilen 1383 und 1384:

```
1383 GOSUB 12900:IF EIN=35 THEN GOSUB 10030
1384 IF EIN>=48 AND EIN<=57 THEN EIN=EIN-31
```

Beachten Sie dabei jedoch, daß ein Open-Befehl mit der Syntax

```
OPEN "O",#nummer,"name"[,satzlänge]
```

nach wie vor nicht erlaubt ist. Darüber hinaus können Sie die Programmzeilen 7130 bis 7137 im Basic-Quelltext löschen, weil diese nie aufgerufen und demzufolge auch nie ausgeführt werden.
Ändern Sie abschließend in Zeile 101 und 102 die Versionsnummer in 1.02, und lassen Sie mit

```
sbasic sbasic
```

Ihr »altes« SBasic 1.01 das neue SBasic 1.02 compilieren. Achten Sie dabei darauf, daß SBasic 1.01 als Exe-Datei und SBasic 1.02 als Bas-Datei im Token-Format vorliegt, andernfalls gibt es eine Fehlermeldung.

(Torsten Harling/zi)

```
[5U1]    1: ;»FILES«
[ZH2]    2: L8D:   MOV   BP,SP
[7B1]    3:        ADD   BP,3
[GB3]    4: ;Stringlänge holen
[C13]    5:        MOV   BL,SS:[BP-1]
[RE1]    6:        MOV   BH,Ø
[PV1]    7:        MOV   DI,BX
[1W2]    8: ;Stringlänge = Ø?
[US1]    9:        AND   BX,BX
[LA1]   1Ø:        JE    X8DØ
[KØ6]   11: ;Muß ein »*.*« angefügt werden?
[9S4]   12:        CMP   BYTE PTR SS:[BP+DI-1],":"
[LA1]   13:        JE    X8DØ
[FV4]   14:        CMP   BYTE PTR SS:[BP+DI-1],"\"
[5G1]   15:        JNE   X8D1
[9H4]   16: ;Ja, dann »*.*« anfügen
[6M2]   17: X8DØ: CMP    BL,252
[AA1]   18:        JA    X8D1
[5H4]   19:        MOV   WORD PTR SS:[BP+DI],"*."
[AS4]   2Ø:        MOV   BYTE PTR SS:[BP+DI+2],"*"
[3B1]   21:        ADD   BL,3
[LM2]   22: X8D1:  CMP   BL,255
[8G1]   23:        JNE   X8D2
[9R1]   24:        MOV   BL,254
[D56]   25: ;String mit Nullbyte abschließen
[6T3]   26: X8D2:  MOV   SS:[BP-1],BL
[WP1]   27:        ADD   BX,BP
[3F1]   28:        MOV   DL,Ø
[XN2]   29:        MOV   SS:[BX],DL
[Ø76]   3Ø: ;Ist eine Laufwerksangabe vorhanden?
```

```
[K44]   31:        CMP   BYTE PTR SS:[BP+1],":"
[BG1]   32:        JNE   X8D3
[ZP2]   33:        MOV   AL,SS:[BP]
[SF5]   34: ;Ja, dann ins Format 1 bis n bringen
[UU1]   35:        AND   AL,ØDFH
[XM1]   36:        SUB   AL,"@"
[LA1]   37:        JB    X8D3
[3M1]   38:        CMP   AL,26
[GA1]   39:        JA    X8D3
[6U1]   4Ø:        MOV   DL,AL
[FN2]   41:        JMP   SHORT X8D4
[9U5]   42: ;Nein, dann Standardlaufwerk holen
[SR2]   43: X8D3:  MOV   AH,19H
[N81]   44:        INT   21H
[6U1]   45:        MOV   DL,AL
[451]   46:        INC   DL
[FC2]   47: X8D4:  PUSH  DX
[A74]   48: ;Dateinummer Ø (»SCRN«)
[3W1]   49:        XOR   BX,BX
[MB6]   5Ø: ;Laufwerksbuchstaben in ASCII wandeln
[3U1]   51:        MOV   AL,DL
[EK1]   52:        ADD   AL,"@"
[172]   53: ;und ausgeben
[AG2]   54:        MOV   DX,XØØ6-LØØ
[D61]   55:        CALL  DX
[RN1]   56:        MOV   AL,":"
[AG2]   57:        MOV   DX,XØØ6-LØØ
[D61]   58:        CALL  DX
[ZT1]   59:        MOV   AL,"\"
[AG2]   6Ø:        MOV   DX,XØØ6-LØØ
```

```
[D61]    61:          CALL   DX
[N71]    62:          POP    DX
[V91]    63:          PUSH   DX
[GHA]    64: ;Platz für momentanes und anzuzeigendes Ver
             zeichnis schaffen
[NR1]    65:          SUB    SP,129
[5Y1]    66:          MOV    SI,SP
[434]    67:          MOV    BYTE PTR SS:[SI],"\"
[X61]    68:          INC    SI
[P91]    69:          PUSH   DS
[SW1]    7Ø:          MOV    AX,SS
[XV1]    71:          MOV    DS,AX
[F85]    72: ;Momentanes Verzeichnis lesen
[3S1]    73:          MOV    AH,47H
[N81]    74:          INT    21H
[4U6]    75: ;In neues Verzeichnis wechseln und lesen
[G13]    76:          MOV    CL,SS:[BP-1]
[VE1]    77:          MOV    CH,Ø
[PU1]    78:          MOV    DI,BP
[SR1]    79:          ADD    DI,CX
[K42]    8Ø: X8D5Ø:   DEC    DI
[314]    81:          CMP    BYTE PTR SS:[DI],"\"
[6H1]    82:          JE     X8D51
[GT1]    83:          LOOP   X8D5Ø
[PS2]    84:          JMP    SHORT X8D53
[AL2]    85: X8D51:   CMP    DI,BP
[TP1]    86:          JNE    X8D52
[151]    87:          INC    DI
[493]    88: X8D52:   PUSH   SS:[DI]
[V91]    89:          PUSH   DX
[5V1]    9Ø:          MOV    DX,BP
[5U3]    91:          MOV    BYTE PTR SS:[DI],Ø
[CT1]    92:          MOV    AH,3BH
[N81]    93:          INT    21H
[N71]    94:          POP    DX
[462]    95:          POP    SS:[DI]
[8H1]    96:          JC     X8D53
[SE5]    97: ;Aktuelles Verzeichnis ausgeben
[XL1]    98:          ADD    SI,64
[3S1]    99:          MOV    AH,47H
[N81]   1ØØ:          INT    21H
[DA5]   1Ø1: ;In altes Verzeichnis wechseln
[YD2]   1Ø2:          LEA    DX,[SI-65]
[CT1]   1Ø3:          MOV    AH,3BH
[N81]   1Ø4:          INT    21H
[782]   1Ø5: X8D53:   POP    DS
[9N3]   1Ø6: ;Verzeichnis ausgeben
[NN3]   1Ø7: X8D6:    MOV    AL,SS:[SI]
[X61]   1Ø8:          INC    SI
[RP1]   1Ø9:          AND    AL,AL
[8B1]   11Ø:          JE     X8D7
[AG2]   111:          MOV    DX,XØØ6-LØØ
[D61]   112:          CALL   DX
[XH1]   113:          JMP    X8D6
[9N2]   114: X8D7:    ADD    SP,129
```

Listing 1. Die Routine »files.asm« verbessert den Files-Befehl in »SBasic«

```
[1CA]     1: ;Funktion »SQR« (Wurzelziehen nach der Zige
             uner-Methode)
[UF2]     2: L92:    MOV    BP,SP
[RZ4]     3: ;Zahl in DI:DX, Exponent in BL
[3F1]     4:         MOV    DL,Ø
[G13]     5:         MOV    DH,SS:[BP+2]
[L13]     6:         MOV    DI,SS:[BP+3]
[K13]     7:         MOV    BL,SS:[BP+5]
[V14]     8: ;Zahl ins Integerformat
[NU1]     9:         OR     DI,8ØØØH
[HØ3]    1Ø: ;Zahl = Ø, dann Ende
[9D1]    11:         CMP    BL,Ø
[ZE1]    12:         JNE    X92Ø
[3PØ]    13:         RET
[HK4]    14: ;Exponent in Integerformat
[NN2]    15: X92Ø:   XOR    BL,8ØH
[HØ4]    16: ;Ist Exponent negativ ?
[PL1]    17:         OR     BL,BL
[XG1]    18:         JNS    X922
[RP1]    19: ;Halbieren
[V41]    2Ø:         NEG    BL
[5E1]    21:         SHR    BL,1
[AG5]    22: ;Muß Mantisse angepaßt werden?
[UE1]    23:         JNC    X921
[343]    24: ;Mantisse anpassen
[AE1]    25:         SHR    DI,1
[8E1]    26:         RCR    DX,1
[NU2]    27: ;Wieder negieren
[B12]    28: X921:   NEG    BL
[9L2]    29:         JMP    SHORT X923
[VN3]    3Ø: ;Exponent ist positiv
[672]    31: X922:   SHR    BL,1
[ØF1]    32:         JNC    X923
[AE1]    33:         SHR    DI,1
[8E1]    34:         RCR    DX,1
[W41]    35:         INC    BL
[778]    36: ;Exponent speichern (Register BX wird gebra
             ucht)
[SN2]    37: X923:   XOR    BL,8ØH
[SØ3]    38:         MOV    SS:[BP+5],BL
[YF8]    39: ;Register DS wird zum Zwischenspeichern ben
             ötigt
```

```
[P91]    4Ø:          PUSH   DS
[DX5]    41: ;24 Bit (plus Rundungsbit) erzeugen
[1P1]    42:          MOV    CX,24
[R56]    43: ;Ergebnis (AX:SI) mit 1 initialisieren
[ZG1]    44:          MOV    SI,1
[WV1]    45:          XOR    AX,AX
[2KA]    46: ;Zwischenwert (BX:BP) initialisieren mit ob
             eren 2 Bit von DI:DX
[KV1]    47:          MOV    BP,AX
[UV1]    48:          MOV    BX,AX
[1E1]    49:          SHL    DX,1
[1D1]    5Ø:          RCL    DI,1
[ØD1]    51:          RCL    BP,1
[1E1]    52:          SHL    DX,1
[1D1]    53:          RCL    DI,1
[ØD1]    54:          RCL    BP,1
[LV6]    55: ;Nun Ergebnis (1) abziehen: a² abziehen
[UT1]    56:          SUB    BP,SI
[FS1]    57:          SBB    BX,AX
[PW4]    58: ;Hole zwei Bits von Mantisse
[C82]    59: X924:    SHL    DX,1
[1D1]    6Ø:          RCL    DI,1
[ØD1]    61:          RCL    BP,1
[8D1]    62:          RCL    BX,1
[1E1]    63:          SHL    DX,1
[1D1]    64:          RCL    DI,1
[ØD1]    65:          RCL    BP,1
[8D1]    66:          RCL    BX,1
[RY1]    67: ;Ergebnis * 2
[EF1]    68:          SHL    SI,1
[4D1]    69:          RCL    AX,1
[ØW1]    7Ø:          MOV    DS,BX
[4V1]    71:          MOV    ES,BP
[P1B]    72: ;Prüfe, ob BX:BP größer ist als 2*a*b + b²
             (mit b=1, und a=AX:SI)
[UT1]    73:          SUB    BP,SI
[FS1]    74:          SBB    BX,AX
[T91]    75:          JC     X925
[UT1]    76:          SUB    BP,SI
[FS1]    77:          SBB    BX,AX
[T91]    78:          JC     X925
[B63]    79: ;b² = 1 abziehen
[GD1]    8Ø:          SUB    BP,1
[WB1]    81:          SBB    BX,Ø
[9F1]    82:          JNC    X926
[RT5]    83: ;BX:BP wiederherstellen (Bit ist Ø)
[CK2]    84: X925:    MOV    BX,DS
[AV1]    85:          MOV    BP,ES
[DL2]    86:          JMP    SHORT X927
[VC3]    87: ;Nächstes Bit ist 1
[B42]    88: X926:    OR     SI,1
[BD2]    89: X927:    LOOP   X924
[XC1]    9Ø: ;Runden
[YC1]    91:          ADD    SI,1
[YA1]    92:          ADC    AX,Ø
[GF5]    93: ;Wert ins richtige Format bringen
[DE1]    94:          SHR    AX,1
[NF1]    95:          RCR    SI,1
[G71]    96:          POP    DS
[742]    97: ;Vorzeichen
[8W1]    98:          MOV    BP,SP
[G13]    99:          MOV    BL,SS:[BP+4]
[9R1]   1ØØ:          AND    AL,7FH
[SM1]   1Ø1:          AND    BL,8ØH
[KL1]   1Ø2:          OR     AL,BL
[783]   1Ø3: ;Ergebnis auf Stack
[LØ3]   1Ø4:          MOV    SS:[BP+4],AL
[813]   1Ø5:          MOV    SS:[BP+2],SI
[3PØ]   1Ø6:          RET
```

Listing 2. Mit »sqr.asm« berechnet »SBasic« die Quadratwurzel schneller und genauer

```
[ZT5]     1: 36 Ø1 8F ØC A4 ØC B7 ØC CA ØC DD ØC FØ ØC Ø3 ØD
[9P5]     2: 16 ØD 21 ØD 2C ØD 37 ØD 42 ØD 4D ØD 5B ØD 6B ØD
[PX5]     3: 7B ØD 81 ØD 87 ØD A8 ØD C6 ØD DF ØD FF ØD 4Ø ØE
[AW5]     4: 49 ØE 78 ØE 8B ØE 9E ØE BA ØE DE ØE 26 ØF 44 ØF
[AC5]     5: CB 1Ø D8 1Ø 1Ø 11 5E 11 6E 11 ØØ 12 ØE 12 1A 12
[XD5]     6: 32 12 6Ø 12 73 12 86 12 99 12 AC 12 BF 12 D2 12
[SD5]     7: FD 12 Ø7 13 49 13 65 13 72 13 98 13 A8 13 BA 13
[XL5]     8: CF 13 E3 13 ØC 14 46 14 6A 14 8C 14 9D 14 DE 14
[TF5]     9: 1A 15 4A 15 DE 15 11 16 44 16 A4 16 D3 16 5F 18
[GN5]    1Ø: 66 18 6B 18 A5 18 AA 18 AF 18 B4 18 EE 18 ØE 19
[UH5]    11: 34 19 39 19 61 19 76 19 8B 19 AØ 19 B5 19 CA 19
[GW5]    12: DF 19 EF 19 Ø8 1A 7A 1A 95 1A A9 1A BB 1A D7 1A
[WY5]    13: EA 1A F6 1A Ø2 1B 1A 1B 2B 1B AF 1C DD 1C EF 1C
[FT5]    14: Ø4 1D 2Ø 1D 9B 1D DØ 1D 21 1E 69 1E 7D 1E 9A 1E
[WW5]    15: DA 1E ØC 1F 18 1F 3C 1F 6A 1F 76 1F 8B 1F DØ 1F
[8P5]    16: E1 1F FE 1F 2C 2Ø B8 2Ø C6 2Ø 93 21 AF 21 D4 21
[SD5]    17: F6 21 18 22 3A 22 45 22 4B 22 6A 22 EA 22 6B 23
[GD5]    18: 79 23 85 23 ØF 26 1B 26 22 26 44 26 4C 28 81 28
[XN5]    19: 11 29 4Ø 29 45 29 F3 29 44 2A A6 2A AB 2A BØ 2A
[BY5]    2Ø: D8 2A 77 2C FF FF 8C DD 8C CA Ø3 C2 8E D8 8E CØ
[KM5]    21: 23 DB 74 11 33 CØ 33 FF B9 Ø8 ØØ F3 AB 8C CØ 4Ø
[BK5]    22: 8E CØ 4B 75 EF 89 2E 2A ØØ C6 Ø6 ØØ ØØ FF C6 Ø6
[AH5]    23: Ø4 ØØ Ø4 C6 Ø6 Ø9 ØØ Ø7 B4 ØF CD 1Ø FE CC 88 26
[9K5]    24: ØB ØØ C6 Ø6 ØD ØØ 17 8C 1E 19 ØØ B4 Ø3 CD 1Ø 89
[9T5]    25: ØE ØF ØØ C7 Ø6 15 ØØ FF FF 8Ø FE 18 72 ØB B8 ØA
[YL5]    26: ØE CD 1Ø B6 17 B4 Ø2 CD 1Ø B9 ØØ 1F B4 Ø1 CD 1Ø
[CN5]    27: BA Ø7 Ø7 B4 ØF CD 1Ø 3C Ø7 75 Ø3 BA ØC ØB 89 16
[YK5]    28: 24 ØØ C7 Ø6 26 ØØ A4 8E C7 Ø6 28 ØØ 1F 7F C7 Ø6
```

Listing 3. Die neue, assemblierte SBasic-Laufzeitbibliothek »sbasic.bin« als Hexdump

```
[Ø95]   29: 2C ØØ 3Ø Ø4 C7 Ø6 2E ØØ 78 Ø7 C3 ØØ ØØ ØØ ØØ ØØ
[ØZ4]   3Ø: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   31: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   32: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   33: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   34: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   35: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   36: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   37: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ZC5]   38: ØØ ØØ ØØ ØØ ØØ ØØ ØØ 5Ø 52 BA FD Ø3 EC DØ E8 73
[FT5]   39: 1F B2 F8 EC 53 2E 8A 1E 1D Ø1 43 83 E3 3F 2E 3A
[5L5]   4Ø: 1E 1E Ø1 74 ØA 2E 88 87 95 ØØ 2E 88 1E 1D Ø1 5B
[1R5]   41: BØ 64 E6 2Ø 5A 58 CF 5Ø 52 BA FD Ø2 EC DØ E8 73
[NT5]   42: 1F B2 F8 EC 53 2E 8A 1E 1F Ø1 43 83 E3 3F 2E 3A
[PK5]   43: 1E 2Ø Ø1 74 ØA 2E 88 87 D5 ØØ 2E 88 1E 1F Ø1 5B
[3S5]   44: BØ 63 E6 2Ø 5A 58 CF 3C ØD 74 Ø9 3C ØA 74 Ø9 FE
[SD5]   45: 47 Ø4 EB Ø4 C6 47 Ø4 ØØ 8A 1F B9 Ø1 ØØ A2 ØE ØØ
[EP5]   46: BA ØE ØØ B4 4Ø CD 21 5B C3 53 8A ØF 8Ø F9 FD 72
[LW5]   47: D6 5Ø 8Ø F9 FE 75 ØE BA FD Ø3 EC A8 2Ø 74 FB B2
[8Z5]   48: F8 58 EE 5B C3 8Ø F9 FD 75 ØE BA FD Ø2 EC A8 2Ø
[4S5]   49: 74 FB B2 F8 58 EE 5B C3 B4 ØF CD 1Ø B4 Ø3 CD 1Ø
[XK5]   5Ø: 59 8A 1E Ø9 ØØ 8Ø F9 Ø7 75 Ø7 B8 Ø7 ØE CD 1Ø 5B
[WR5]   51: C3 8Ø F9 Ø9 75 2E E8 CC ØØ B4 Ø3 CD 1Ø 8A C2 24
[GD5]   52: Ø7 B9 Ø8 ØØ 2A C8 B8 2Ø Ø9 CD 1Ø C6 Ø6 ØA ØØ ØØ
[CM5]   53: Ø2 D1 3A 16 ØB ØØ 76 Ø6 FE CA FE ØE ØA ØØ B4 Ø2
[YM5]   54: CD 1Ø 5B C3 8Ø F9 ØA 74 Ø5 8Ø F9 ØD 75 Ø5 E8 A1
[CG5]   55: ØØ 5B C3 8Ø F9 ØB 75 11 B2 ØØ 8A 36 ØC ØØ B4 Ø2
[SM5]   56: CD 1Ø C6 Ø6 ØA ØØ ØØ 5B C3 8Ø F9 ØC 75 Ø8 B8 ØØ
[UL5]   57: Ø6 E8 9C ØØ EB E2 8Ø F9 1C 75 12 B4 Ø3 CD 1Ø 3A
[BR5]   58: 16 ØB ØØ 74 Ø4 FE C2 EB 15 B2 ØØ EB Ø9 8Ø F9 1F
[7M5]   59: 75 17 B4 Ø3 CD 1Ø 3A 36 ØD ØØ 74 ØB FE C6 B4 Ø2
[YM5]   6Ø: CD 1Ø C6 Ø6 ØA ØØ ØØ 5B C3 8Ø F9 1D 75 13 B4 Ø3
[PW5]   61: CD 1Ø 8Ø FA ØØ 74 Ø4 FE CA EB E3 8A 16 ØB ØØ EB
[CN5]   62: Ø9 8Ø F9 1E 75 ØE B4 Ø3 CD 1Ø 3A 36 ØC ØØ 74 D7
[ZT5]   63: FE CE EB CA E8 ØE ØØ 8A C1 B4 Ø9 B9 Ø1 ØØ CD 1Ø
[2E5]   64: E8 46 ØØ 5B C3 3A 16 ØB ØØ 75 3D 8Ø 3E ØA ØØ ØØ
[NG5]   65: 74 36 51 B2 ØØ 3A 36 ØD ØØ 75 Ø8 B8 Ø1 Ø6 E8 ØF
[VK5]   66: ØØ EB Ø2 FE C6 B4 Ø2 CD 1Ø C6 Ø6 ØA ØØ ØØ 59 C3
[RM5]   67: 53 51 52 8A FB B1 ØØ 8A 2E ØC ØØ 8A 16 ØB ØØ 8A
[MM5]   68: 36 ØD ØØ CD 1Ø 5A 59 5B C3 B4 Ø3 CD 1Ø 3A 16 ØB
[ML5]   69: ØØ 74 ØC FE C2 B4 Ø2 CD 1Ø C6 Ø6 ØA ØØ ØØ C3 C6
[RN5]   7Ø: Ø6 ØA ØØ FF C3 BØ 1A 8B 3E 17 ØØ 23 FF 74 ØF 2E
[BN5]   71: 8A Ø5 47 3C 1A 75 Ø3 2E 8B 3D 89 3E 17 ØØ C3 23
[1U5]   72: DB 78 E2 8A Ø7 3C FF 74 71 53 3C FE 75 2Ø 2E 8A
[SN5]   73: 1E 1E Ø1 2E 3A 1E 1D Ø1 74 F4 FE C3 8Ø E3 3F 2E
[UN5]   74: 8A 87 95 ØØ 2E 88 1E 1E Ø1 5B 88 7F 1Ø C3 3C FD
[2M5]   75: 75 2Ø 2E 8A 1E 2Ø Ø1 2E 3A 1E 1F Ø1 74 F4 FE C3
[GL5]   76: 8Ø E3 3F 2E 8A 87 D5 ØØ 2E 88 1E 2Ø Ø1 5B 88 7F
[DM5]   77: 1Ø C3 5B 8A 4F 1Ø 51 8Ø F9 1A 74 17 53 8A D8 B9
[RL5]   78: Ø1 ØØ BA ØE ØØ B4 3F CD 21 5B 72 Ø7 23 CØ AØ ØE
[3H5]   79: ØØ 75 Ø2 BØ 1A 88 47 1Ø 58 C3 AØ ØE ØØ 3C FF 75
[CK5]   8Ø: Ø5 B4 Ø8 CD 21 C3 83 3E 15 ØØ FF 74 28 3C ØØ 75
[SL5]   81: Ø3 BØ 1A C3 FE ØE ØE ØØ 53 B4 ØF CD 1Ø B4 Ø8 CD
[XG5]   82: 1Ø 5Ø B4 Ø3 CD 1Ø 3A 16 ØB ØØ 73 Ø6 FE C2 B4 Ø2
[FT5]   83: CD 1Ø 58 5B C3 53 B4 ØF CD 1Ø B4 Ø3 CD 1Ø 89 ØE
[KF5]   84: 15 ØØ 52 33 FF 3A 16 ØB ØØ 76 Ø4 8A 16 ØB ØØ B4
[TK5]   85: Ø2 CD 1Ø 8B ØE 24 ØØ 23 FF 74 Ø2 B5 Ø4 B4 Ø1 CD
[EG5]   86: 1Ø B4 Ø8 CD 21 5Ø B9 ØØ 1F B4 Ø1 CD 1Ø 58 3C 2Ø
[CN5]   87: 72 44 8A 1E Ø9 ØØ 23 FF 74 31 5Ø 8A CA 8A 16 ØB
[LS5]   88: ØØ FE CA 78 1F 3A D1 72 1B B4 Ø2 CD 1Ø B4 Ø8 CD
[2H5]   89: 1Ø FE C2 B4 Ø2 CD 1Ø 51 B9 Ø1 ØØ B4 Ø9 CD 1Ø 59
[BV5]   9Ø: FE CA EB DD 8A D1 B4 Ø2 CD 1Ø 58 B9 Ø1 ØØ B4 Ø9
[KT5]   91: CD 1Ø FE C2 EB 8F 3C ØØ 75 4D B4 Ø8 CD 21 3C 4B
[TP5]   92: 75 ØB 58 5Ø 3A DØ 76 Ø2 FE CA E9 78 FF 3C 4D 75
[LU5]   93: Ø4 FE C2 EB F5 3C 47 75 Ø4 5A 52 EB ED 3C 4F 75
[UH5]   94: 1A 8A 16 ØB ØØ 59 51 B4 Ø2 CD 1Ø B4 Ø8 CD 1Ø 3C
[RW5]   95: 2Ø 75 DE 3A D1 74 D3 FE CA EB EC 3C 53 74 14 3C
[TR5]   96: 52 75 C7 F7 D7 EB C3 3C Ø8 75 3E 59 51 3A D1 76
[2L5]   97: 35 FE CA 52 8A 1E Ø9 ØØ 3A 16 ØB ØØ 73 1B FE C2
[6P5]   98: B4 Ø2 CD 1Ø B4 Ø8 CD 1Ø FE CA B4 Ø2 CD 1Ø B9 Ø1
[2S5]   99: ØØ B4 Ø9 CD 1Ø FE C2 EB DF B4 Ø2 CD 1Ø B8 2Ø Ø9
[ZR5]   9ØØ: B9 Ø1 ØØ CD 1Ø 5A E9 FC FE 3C ØD 74 Ø7 3C ØA 74
[8R5]   9Ø1: F5 E9 1E FF 59 8A 16 ØB ØØ B4 Ø2 CD 1Ø B4 Ø8 CD
[4S5]   9Ø2: 1Ø 3C 2Ø 75 Ø8 FE CA 78 Ø4 3A D1 73 EC FE C2 2A
[LP5]   9Ø3: D1 88 16 ØE ØØ 8A D1 B4 Ø2 CD 1Ø 5B E9 7B FE 8Ø
[NN5]   9Ø4: 3F FF 75 13 BØ ØD E8 6Ø FC 8B ØE 15 ØØ B4 Ø1 CD
[AL5]   9Ø5: 1Ø C7 Ø6 15 ØØ FF FF C2 Ø2 ØØ 8B EC 83 C5 Ø4 8A
[FM5]   9Ø6: 8E ØØ Ø1 8A 6E ØØ 33 FF E3 1C 8Ø F9 ØØ 74 14 8Ø
[6P5]   9Ø7: FD ØØ 74 15 47 FE C9 FE CD 8A 83 ØØ Ø1 3A Ø3 74
[GH5]   9Ø8: E7 77 Ø6 BØ Ø2 C3 BØ Ø1 C3 BØ Ø4 C3 8B EC B2 ØØ
[4N5]   9Ø9: 8A 76 Ø6 8B 76 Ø7 8A 5E Ø9 BØ ØØ 8A 66 Ø2 8B 7E
[1N5]   91Ø: Ø3 8A 7E Ø5 3A DF 73 Ø5 92 87 F7 86 DF 8Ø FF ØØ
[ØR5]   911: 74 6A 8A CB 2A CF 8Ø F9 18 77 61 B5 ØØ 56 33 F7
[8S5]   912: 9C 8B 76 FE 81 CE ØØ 8Ø 81 CF ØØ 8Ø E3 Ø6 D1 EF
[WS5]   913: D1 D8 E2 FA 9D 78 ØE Ø3 DØ 13 F7 73 39 D1 DE D1
[9T5]   914: DA FE C3 EB 31 2B DØ 1B F7 72 Ø8 75 14 23 D2 75
[SN5]   915: 1Ø EB 1A F7 D2 83 C2 Ø1 F7 D6 83 D6 ØØ 8Ø 76 FF
[HR5]   916: 8Ø 23 F6 78 11 D1 E2 D1 D6 FE CB 75 F4 33 F6 33
[SM5]   917: D2 B3 ØØ 58 EB Ø6 58 ØD FF 7F 23 FØ 88 76 Ø6 89
[BL5]   918: 76 Ø7 88 5E Ø9 C2 Ø4 ØØ 8B EC 8Ø 7E Ø9 ØØ 74 5B
[HL5]   919: 8Ø 7E Ø5 ØØ 74 55 8A 5E Ø2 8A 7E Ø6 8A C3 F6 E7
[9N5]   92Ø: 8B F8 8B 76 Ø3 8B 6E Ø7 8B C6 33 C5 ØD FF 7F 5Ø
[ZM5]   921: 81 CE ØØ 8Ø 81 CD ØØ 8Ø 8B C6 F7 E5 8B C8 87 F2
[ML5]   922: BØ ØØ 8A E7 F7 E2 Ø3 F8 13 CA 83 D6 ØØ BØ ØØ 8A
[8R5]   923: E3 F7 E5 Ø3 F8 13 CA 83 D6 ØØ 58 8B EC 8A 5E Ø5
[6F5]   924: 8A 7E Ø9 81 F3 8Ø 8Ø Ø2 DF 71 ØB 33 CØ 89 46 Ø6
[DL5]   925: 89 46 Ø8 C2 Ø4 ØØ 8Ø F3 8Ø 23 F6 78 ØA D1 E7 D1
[PN5]   926: D1 D1 D6 FE CB 75 F2 81 C1 8Ø ØØ 83 D6 ØØ 73 Ø6
[YS5]   927: D1 DE D1 D9 FE C3 23 FØ 88 6E Ø6 89 76 Ø7 88 5E
[EE5]   928: Ø9 C2 Ø4 ØØ 8B EC 8Ø 7E Ø9 ØØ 74 76 8Ø 7E Ø5 ØØ
[TG5]   929: 74 73 8B 76 Ø6 8A 5E Ø8 B7 ØØ 8B 7E Ø2 8A 56 Ø4
[PK5]   93Ø: B6 ØØ 8A C3 32 C2 ØC 7F 5Ø 8Ø CB 8Ø 8Ø CA 8Ø B1
[BR5]   931: 19 33 ED BØ ØØ 2B F7 1B DA 73 Ø4 Ø3 F7 13 DA F5
[HT5]   932: D1 D5 DØ DØ D1 E6 D1 D3 FE C9 22 CØ 79 E7 2B F7
[ER5]   933: 1B DA F5 83 D5 ØØ 14 ØØ 73 Ø6 DØ D8 D1 DD FE C1
[5R5]   934: 5B 8B D5 8B EC 22 C3 89 56 Ø6 88 46 Ø8 8A 46 Ø9
[NC5]   935: 8A 66 Ø5 35 8Ø 8Ø Ø2 C1 2A C4 7Ø Ø9 35 8Ø ØØ 88
[1D5]   936: 46 Ø9 C2 Ø4 ØØ 33 CØ 89 46 Ø6 89 46 Ø8 C2 Ø4 ØØ
[LM5]   937: 8B EC 8A 5E Ø5 22 DB 74 Ø2 8Ø BØ 8Ø 84 46 Ø4 74 Ø9
[BB5]   938: 33 CØ 89 46 Ø2 89 46 Ø4 C3 88 46 Ø5 81 7E Ø4 35
[FK5]   939: 8Ø 75 Ø5 81 7E Ø2 F4 Ø4 73 Ø5 FE 46 Ø5 FE CB 32
[NL5]   94Ø: FF 53 FF 76 Ø4 FF 76 Ø2 B8 8Ø 81 5Ø 33 CØ 5Ø E8
[LP5]   941: FA FD FF 76 1Ø FF 76 ØE B8 ØØ 81 5Ø 33 CØ 5Ø E8
[8R5]   942: EA FD E8 1F FF 8B 46 Ø8 89 46 1Ø 5Ø 8B 46 Ø6 89
[GH5]   943: 46 ØE 5Ø E8 72 FE 8B 46 Ø8 8B 5E ØA 53 5Ø 53 5Ø
[1Z5]   944: E8 65 FE B8 5E 7F 5Ø B8 4D 57 5Ø E8 BE FD FF 76
[2S5]   945: ØC FF 76 ØA E8 51 FE B8 13 8Ø 5Ø B8 Ø8 9B 5Ø E8
```

```
[HX5]   146: AA FD FF 76 ØC FF 76 ØA E8 3D FE B8 76 8Ø 5Ø B8
[ZS5]   147: 93 38 5Ø E8 96 FD E8 2F FE B8 38 82 5Ø B8 3B AA
[8Y5]   148: 5Ø E8 88 FD FF 76 1Ø FF 76 ØE E8 1B FE 8A 46 ØA
[ØL5]   149: B4 ØØ 34 8Ø 74 17 8A D8 79 Ø2 F6 D8 B4 88 22 CØ
[8S5]   15Ø: 78 Ø6 DØ EØ FE CC EB F6 8Ø CB 7F 22 C3 5Ø 33 CØ
[CR5]   151: 5Ø E8 58 FD B8 31 8Ø 5Ø B8 18 72 5Ø E8 E9 FD 8F
[ZM5]   152: 46 ØE 8F 46 1Ø 58 C3 8B EC B2 ØØ 8B 76 Ø6 8B 46
[6P5]   153: Ø8 8B 7E Ø2 8B 5E Ø4 8A C8 8A EB 32 E9 78 19 ØC
[BK5]   154: 8Ø 8Ø CB 8Ø 3B C3 75 Ø4 3B F7 74 ØB 77 Ø3 8Ø F1
[AT5]   155: 8Ø DØ E1 72 Ø7 FE C2 C3 DØ EØ 73 F9 FE CA C3 53
[ZS5]   156: 8A 1F 8Ø FB FF 74 4F 8Ø FB Ø5 72 4A 33 CØ 8E CØ
[XR5]   157: 8Ø FB FE 75 1C BA F9 Ø3 EE E4 21 ØC 1Ø E6 21 2E
[3E5]   158: A1 15 Ø1 26 A3 3Ø ØØ 2E A1 17 Ø1 26 A3 32 ØØ EB
[EU5]   159: 25 8Ø FB FD 75 1C BA F9 Ø2 EE E4 21 ØC Ø8 E6 21
[9A5]   16Ø: 2E A1 19 Ø1 26 A3 2C ØØ 2E A1 1B Ø1 26 A3 2E ØØ
[AR5]   161: EB Ø4 B4 3E CD 21 5B C3 Ø7 B8 38 81 5Ø B8 3B AA
[5L5]   162: 5Ø E8 44 FD 33 D2 33 CØ 8B 5E Ø8 8Ø FF 81 72 24
[VL5]   163: 8A E3 8Ø CC 8Ø 8A 46 Ø7 8A 56 Ø6 8Ø FF 9Ø 73 Ø9
[MT5]   164: D1 E8 8Ø D6 ØØ FE C7 EB F2 22 DB 79 Ø7 F7 D8 23
[AH5]   165: D2 74 Ø1 48 5Ø FF 76 Ø8 FF 76 Ø6 B7 ØØ 23 CØ 74
[DW5]   166: ØC 79 Ø2 F7 D8 B7 9Ø FE CF D1 EØ 79 FA 8Ø E3 8Ø
[PK5]   167: 8Ø E4 7F ØA E3 8A DC 8A EØ BØ ØØ 8Ø F3 8Ø 53 5Ø
[BU5]   168: E8 49 FC B8 34 71 5Ø B8 3C 58 5Ø FF 76 Ø8 FF 76
[YV5]   169: Ø6 E8 D4 FC B8 16 74 5Ø B8 B7 7E 5Ø E8 2D FC FF
[3W5]   17Ø: 76 ØC FF 76 ØA E8 CØ FC B8 2F 77 5Ø B8 DF EE 5Ø
[TZ5]   171: E8 19 FC FF 76 ØC FF 76 ØA E8 AC FC B8 1D 7A 5Ø
[2X5]   172: B8 17 84 5Ø E8 Ø5 FC FF 76 ØC FF 76 ØA E8 98 FC
[4T5]   173: B8 63 7C 5Ø B8 59 59 5Ø E8 F1 FB FF 76 ØC FF 76
[RY5]   174: ØA E8 84 FC B8 75 7E 5Ø B8 E9 FD 5Ø E8 DD FB FF
[DN5]   175: 76 ØC FF 76 ØA E8 7Ø FC B8 31 8Ø 5Ø B8 13 72 5Ø
[5X5]   176: E8 C9 FB FF 76 ØC FF 76 ØA E8 5C FC B8 ØØ 81 5Ø
[AH5]   177: 33 CØ 5Ø E8 B6 FB 58 5B 59 59 59 Ø2 F9 75 Ø4 33
[CP5]   178: CØ 33 DB 89 46 1Ø 89 5E 12 Ø6 C3 8B EC 8B 46 Ø4
[PM5]   179: 8Ø FC 98 73 5Ø 22 CØ 79 1C FF 76 Ø4 FF 76 Ø2 B8
[UW5]   18Ø: FF 8Ø 5Ø B8 FF FF 5Ø E8 82 FB 8B EC 83 C5 Ø4 8F
[CR5]   181: 46 Ø2 8F 46 Ø4 B8 FF FF B3 FF B1 9Ø 2A 4E Ø5 73
[XG5]   182: Ø7 8Ø C1 Ø8 D2 EØ EB 17 BØ ØØ 8Ø E9 Ø8 73 Ø7 8Ø
[DL5]   183: C1 Ø8 D2 E4 EB Ø9 B4 ØØ 8Ø F9 Ø8 73 Ø9 D2 E3 21
[YE5]   184: 46 Ø2 2Ø 5E Ø4 C3 89 46 Ø2 89 46 Ø4 C3 Ø7 B8 49
[GR5]   185: 81 5Ø B8 DA ØF 5Ø E8 33 FB Ø6 Ø7 B8 22 7E 5Ø B8
[3X5]   186: 81 F9 5Ø E8 C2 FB 8B EC FF 76 Ø2 FF 76 ØØ E8 7A
[1M5]   187: FF 8B EC 8Ø 7E Ø3 ØØ 74 Ø4 8Ø 76 Ø2 8Ø 8E ØC FB
[XE5]   188: 8B EC 8Ø 7E Ø3 ØØ 74 Ø4 8Ø 76 Ø2 8Ø B8 ØØ 7F 5Ø
[DM5]   189: 33 CØ 5Ø E8 F6 FA 8B EC F6 46 Ø2 8Ø 74 19 B8 ØØ
[PN5]   19Ø: 8Ø 5Ø 33 CØ 5Ø E8 E4 FA 8B EC 8Ø 66 Ø2 7F B8 8Ø
[LC5]   191: 7F 5Ø 33 CØ 5Ø EB 11 8Ø 7E Ø3 ØØ 74 Ø4 8Ø 76 Ø2
[WP5]   192: 8Ø B8 ØØ 7F 5Ø 33 CØ 5Ø E8 C1 FA 8B EC FF 76 Ø2
[WS5]   193: FF 76 ØØ FF 76 Ø2 FF 76 ØØ E8 4C FB B8 E6 84 5Ø
[3T5]   194: B8 2D 1A 5Ø 8B EC FF 76 Ø6 FF 76 Ø4 E8 39 FB B8
[BW5]   195: 28 86 5Ø B8 FB Ø7 5Ø E8 92 FA 8B EC FF 76 Ø6 FF
[AP5]   196: 76 Ø4 E8 23 FB B8 99 87 5Ø B8 99 68 5Ø E8 7C FA
[TT5]   197: 8B EC FF 76 Ø6 FF 76 Ø4 E8 ØD FB B8 23 87 5Ø B8
[ØZ5]   198: DF 35 5Ø E8 66 FA 8B EC FF 76 Ø6 FF 76 Ø4 E8 F7
[5X5]   199: FA B8 A5 86 5Ø B8 E7 5D 5Ø E8 5Ø FA 8B EC FF 76
[LS5]   2ØØ: Ø6 FF 76 Ø4 E8 E1 FA B8 49 83 5Ø B8 DA ØF 5Ø E8
[ZØ6]   2Ø1: 3A FA 8B EC 8F 46 Ø4 8F 46 Ø6 E8 CB FA Ø6 C3 BB
[YP5]   2Ø2: Ø3 ØØ E8 1A FD 4B 75 FA 8B ØE ØF ØØ B4 Ø1 CD 1Ø
[PL5]   2Ø3: B4 4C CD 21 8B EC 33 D2 8B 46 Ø4 3B 46 Ø2 7E Ø1
[HL5]   2Ø4: 4A 89 56 Ø4 C2 Ø2 ØØ 8B EC 33 D2 8B 46 Ø4 3B 46
[7H5]   2Ø5: Ø2 7D Ø1 4A 89 56 Ø4 C2 Ø2 ØØ 8B EC 33 D2 8B 46
[FE5]   2Ø6: Ø4 3B 46 Ø2 75 Ø1 4A 89 56 Ø4 C2 Ø2 ØØ 8B EC 33
[DD5]   2Ø7: D2 8B 46 Ø4 3B 46 Ø2 7C Ø1 4A 89 56 Ø4 C2 Ø2 ØØ
[KH5]   2Ø8: 8B EC 33 D2 8B 46 Ø4 3B 46 Ø2 7F Ø1 4A 89 56 Ø4
[SK5]   2Ø9: C2 Ø2 ØØ 8B EC 33 D2 8B 46 Ø4 3B 46 Ø2 74 Ø1 4A
[SF5]   21Ø: 89 56 Ø4 C2 Ø2 ØØ 8B EC 8B 46 Ø2 Ø1 46 Ø4 C2 Ø2
[2H5]   211: ØØ 8B EC 8B 46 Ø2 29 46 Ø4 C2 Ø2 ØØ 8B EC 8B 46
[XD5]   212: Ø2 21 46 Ø4 C2 Ø2 ØØ 8B EC 8B 46 Ø2 Ø9 46 Ø4 C2
[LL5]   213: Ø2 ØØ 8B EC 8B 46 Ø2 31 46 Ø4 C2 Ø2 ØØ 8B EC 8B
[PG5]   214: 46 Ø4 F7 66 Ø2 89 46 Ø4 C2 Ø2 ØØ 8B EC 8B 46 Ø4
[UK5]   215: 33 D2 F7 7E Ø2 89 46 Ø4 C2 Ø2 ØØ 8B EC 8B 46 Ø4
[KK5]   216: 33 D2 F7 7E Ø2 89 56 Ø4 C2 Ø2 ØØ 8B EC F7 56 Ø2
[HX5]   217: C3 8B EC F7 5E Ø2 C3 8B EC 8B 5E ØØ 8C DA C5 76
[416]   218: Ø2 81 EC FA ØØ 8B FC 8C DØ 8E CØ FC AC AA 8A C8
[7V5]   219: B5 ØØ F3 A4 8E DA FF E3 8B F4 83 C6 Ø2 1E 8C DØ
[AT5]   22Ø: 8E D8 33 FF 36 8E 84 ØØ Ø1 FC AC AA 8A C8 B5 ØØ
[PN5]   221: F3 A4 1F C2 Ø2 Ø1 8B EC 8B 5E Ø2 BØ ØD BA A3 Ø1
[8W5]   222: FF D2 8Ø 3F FF 74 Ø7 BØ ØA BA A3 Ø1 FF D2 C3 8B
[HH5]   223: EC 83 C5 Ø2 8B 9E ØØ Ø1 8A 4E ØØ B5 ØØ E3 ØD 45
[1M5]   224: 8A 46 ØØ 51 BA A3 Ø1 FF D2 59 E2 F3 C2 ØØ Ø1 81
[XR5]   225: EC FC ØØ 8B EC 8B 86 FE ØØ B3 2Ø 23 CØ 79 Ø4 F7
[DN5]   226: D8 B3 2D 88 5E Ø1 BB ØA ØØ 33 C9 33 D2 F7 F3 8Ø
[7S5]   227: CA 3Ø 52 23 CØ EØ F4 F7 D9 8A D9 BF Ø2 ØØ 58 88
[TS5]   228: Ø3 47 E2 FA C6 Ø3 2Ø FE C3 88 5E ØØ FF A6 FC ØØ
[TU5]   229: 33 DB BØ ØC BA A3 Ø1 FF E2 8B F4 83 C6 Ø2 1E 8C
[SS5]   23Ø: DØ 8E D8 8E CØ 8A 8C ØØ Ø1 B5 Ø1 8B F9 AC Ø3 FE
[7N5]   231: B5 ØØ F7 D9 81 C1 FF ØØ 3A C1 73 Ø2 8A C8 ØØ 8C
[HP5]   232: FF ØØ F3 A4 1F C2 ØØ Ø1 81 EC FC ØØ 8B EC BØ Ø1
[XT5]   233: 8A A6 FE ØØ 89 46 ØØ FF A6 FC ØØ 8B EC 33 D2 39
[DL5]   234: 56 Ø2 74 Ø6 7F Ø3 4A EB Ø1 42 89 56 Ø2 C3 8B EC
[PL5]   235: B4 ØF CD 1Ø 8A 56 Ø2 8A 76 Ø4 81 EA Ø1 Ø1 B4 Ø2
[XM5]   236: CD 1Ø C6 Ø6 ØA ØØ ØØ C2 Ø4 ØØ 8B EC 8A 46 Ø4 8A
[9N5]   237: 66 Ø2 B1 Ø4 D2 E4 8Ø E4 7Ø 8A D8 24 ØF ØA C4 B1
[CH5]   238: Ø3 D2 E3 8Ø E3 8Ø ØA C3 A2 Ø9 ØØ C2 Ø4 ØØ B4 ØF
[2R5]   239: CD 1Ø B4 Ø3 CD 1Ø 8B EC 8A 5E Ø2 8Ø FB 28 75 2Ø
[WG5]   24Ø: 8Ø FC 28 74 14 3C Ø6 74 Ø4 BØ Ø1 EB Ø2 BØ Ø5 FE
[TG5]   241: CB 88 1E ØB ØØ B4 ØØ CD 1Ø B4 Ø1 CD 1Ø C2 Ø2 ØØ
[YG5]   242: 8Ø FB 5Ø 75 F4 8Ø FC 5Ø 74 EF 3C Ø4 73 Ø4 BØ Ø3
[KV5]   243: EB DD BØ Ø6 EB D9 8B EC 8A 46 Ø4 FE C8 A2 ØC ØØ
[KS5]   244: 8A 46 Ø2 FE C8 A2 ØD ØØ 33 DB BØ ØB BA A3 Ø1 FF
[FR5]   245: D2 C2 Ø4 ØØ 8B F4 83 C6 Ø8 8B DE 1E 8C DØ 8E D8
[TM5]   246: AC B4 ØØ 8B D6 Ø3 FØ 88 24 8Ø 7F Ø5 3A 75 18 8B
[NS5]   247: 47 Ø1 25 DF DF 8B 4F Ø3 81 E1 DF 3D 53 43 75
[3U5]   248: ØE 81 F9 52 4E 74 Ø3 E9 C2 ØØ BØ FF E9 DC ØØ 3D
[RM5]   249: 4B 59 75 Ø8 81 F9 42 44 74 FØ EB EB 3D 43 4F 75
[DL5]   25Ø: E6 B6 Ø3 81 F9 4D 11 74 Ø8 FE CE 81 F9 4D 12 75
[ØP5]   251: D6 B2 FB BØ 83 EE 8D 77 Ø6 52 33 C9 AC 2C 3Ø 72
[7N5]   252: 12 3C ØA 73 ØE B4 ØØ 8B F8 BØ ØA F7 E1 Ø3 C7 8B
[5L5]   253: C8 EB E9 23 C9 75 Ø3 B9 2C Ø1 B8 ØØ C2 BA Ø1 ØØ
[BP5]   254: F7 F1 5A B2 F8 EE 42 8A C4 EE 42 42 BØ Ø3 EE 42
[AU5]   255: BØ ØB EE B2 F9 BØ Ø1 EE 4A EC 1E 33 CØ 8E D8 B1
[MH5]   256: FE 8Ø FE Ø3 75 1F 2E A3 1D Ø1 B8 21 Ø1 87 Ø6 3Ø
[ØH5]   257: ØØ 2E A3 15 Ø1 8C C8 87 Ø6 32 ØØ 2E A3 17 Ø1 E4
[KS5]   258: 21 24 EF EB 1F FE C9 2E A3 1F Ø1 B8 51 Ø1 87 Ø6
```

Listing 3. Die neue, assemblierte SBasic-Laufzeitbibliothek »sbasic.bin« als Hexdump

DMV-Versandservice

Die preiswerte Alternative für aktive PC-Anwender!

Textverarbeitung

ConText – Textverarbeitung mit Konzept
kompromißlose WYSIWYG-Darstellung im Grafikmodus
einfache und blitzschnelle Bedienung
Vielseitigkeit, hohes Tempo und Flexibilität

ConText PRO (M)(D)	DM 199,–
ConText EASY	DM 99,–
Upgrade auf ConText PRO	DM 129,–

Zusatzprogramme:

Adreßverwaltung ConText-ADREVA	DM 49,–

PCpur-Software

MultiFont Plus
Zeichengenerator für Matrixdrucker
Zusatzfonts MultiFont Plus DM 69,– / DM 39,–

StCII-Tiger
Programmierbarer Editor DM 69,–
Festplatten-Optimierer DM 69,–

Anwendungen

ARGO 3.0
GEM-Benutzeroberfläche (D) DM 199,–
Update auf Version 3.0 DM 69,–

Hyperkey Plus 4.0
PC/XT/AT-Tastaturbelegung (M) DM 99,–
Update auf Version 4.0 DM 29,–

DMV-Power-Pack
Viele Grundausstattung für Einsteiger und Profis
ARGO 3.0
ConText PRO
ConText ADREVA DM 349,–

CirrusDoktor
Abwehr und Schutz vor Computerviren DM 99,–

MaskEdit Plus
Maskengenerator

MaskEdit Plus I für TTurbo Pascal 3/4/5 und dBase III + (M)(D)	DM 99,–
MaskEdit Plus II für TTurbo C und Microsoft C (M)(D)	DM 99,–
MaskEdit Plus III für TTopSpeed Modula-2 (M)(D)	DM 99,–
Paket MaskEdit Plus I + II, (M)(D)	DM 149,–
Paket MaskEdit Plus I bis III, (M)(D)	DM 199,–

Software-Entwicklung

StCII DisAsm
intelligenter MS-DOS-Disassembler DM 199,–

MAI.-Tech
Hochsprachen-Assembler-Interface DM 199,–

TopSpeed Modula-2
Der Schritt in die Zukunft: TopSpeed
Derzeitige deutsche Compilerversion
TopSpeed Compiler
Dateiverwaltung + Maskengenerator DM 348,–
TopSpeed TechKit DM 248,–
TopSpeed Debugger DM 248,–
Sonderpreis für Compiler und Debugger DM 555,–
OS/2-Version DM 698,–

DOS-Software-Tools

RSM – Resident-Software-Manager (M)	DM 69,–
PC-Utilities	DM 69,–
MS-DOS-Utilities	DM 49,–
3D-Draw (D)	DM 99,–
GW-Basic-Toolboxen	
Oberfläche	DM 69,–
DOS-Funktionen	DM 69,–
Hercules-Grafik	DM 69,–
Navigator für Microsoft Flugsimulator III	DM 69,–

TOOLBOX-Spezial

I: Turbo-Prolog-Toolbox
II: PasComp – Standard Pascal Compiler, TP 4/5
III: Echtzeitverarbeitungs-System, TP 3
IV: Formelcompiler CALC & Kurvendiskussion, TP 3/4/5
V: STRUKTO Plus "Pascal"- Struktogrammgenerator
VI: Knobelei und Unterhaltung & Sprite-Editor, TP 4/5
VII: Deutsche Silbentrennung - Algorithmen & Programm
VIII: Spieltheorie – Kalaha, "Vier gewinnt" und Schach
IX: HPGL-Plotter-Toolbox mit 3D-Option, TP 3/4/5
X: CRC-Softwaresicherung, Anwendung und Toolbox
XI: MAP, MARK & RELEASE, TSR-Handling, TP 4/5
XIII: Das Beste aus der Trickkiste der TOOLBOX Vol.I
XIV: Das Beste aus der Trickkiste der TOOLBOX Vol.II
XV: Fitting, num. Integration & Lineare Gl.-Sys.
XVI: Natürlichsprachliche Schnittstelle

Alle TOOLBOX-Spezial (D) jeweils nur DM 35,–

Vertriebsprodukte

C-Funktions-Sammlung
für Turbo C und Microsoft C DM 99,–

GEMInterface PC
Toolkit zur Erstellung eigener GEM-Applikationen, für
Turbo Pascal 4/5 DM 198,–
TopSpeed Modula-2 DM 198,–

Bücher

Das ausverkaufte DOS Extra I jetzt als Taschenbuch!

DOS-Taschenbuch Band 1 "Von Assembler bis MS-DOS"	DM 29,80
Databox Band I Alle Listings und Beispiele auf Datenträger	DM 24,–

Exklusiv für alle Amstrad/Schneider-Anwender

Die Basic-2-Toolbox	DM 49,–
Diskette zum Buch	DM 29,–

Unterhaltung

Fraktal Generator 3-D	DM 69,–
Systemanforderungen: EGA-/VGA-Karte oder Amstrad/Schneider PC 1512, (D)	
PC-Spielebox No.1 BIMBO II, BOUNCER	DM 49,–
PC-Spielebox No.2 JAKEY, STARVISION	DM 49,–
Know PC	DM 49,–
Know-PC-Ergänzungsfragen	DM 29,–

Sonderhefte

DOS Extra III	DM 18,–
DOS Extra IV	DM 18,–
DOS Extra V	DM 18,–
DOS Extra VI	DM 18,–
DOS Extra VII	DM 18,–
DOS Extra VIII	DM 18,–
DOS Extra IX	DM 18,–
PASCAL Sonderdruck Das Beste aus PASCAL 11'86 bis 6'87	DM 25,–
DMV-Computer-Wissen I Basic-Toolbox	DM 18,–

Zeitschriften

Paketangebote

PC Amstrad Superpack 1/87 bis 12/87 + 3x86 + 2 Sammelordner
Jeweils 6 PASCAL-Ausgaben freier Wahl,
Pack II: 6 Hefte, PASCAL 7'87 bis 12'87
Pack III: 7 Hefte, PASCAL 1'88 bis 8/9'88

Für die DATABOX-Angebote zu den Paketen beachten Sie bitte die entsprechende Werbung in jeder Ausgabe.

Sammelmappen

für DOS International, PC Amstrad International, PCpur und toolbox je 2 Stück DM 15,80

Gibt es wieder Gratis-Chips bei DMV?

HANNOVER MESSE CeBIT'90
Welt-Centrum Büro · Information · Telekommunikation
21. – 28. MÄRZ 1990
Wir stellen aus: Halle 7, Stand E50

Unabhängig von der Anzahl der bestellten Produkte, Programme und Bücher berechnen wir für das Inland DM 4,– bzw. für das Ausland DM 6,– Porto und Verpackung.

Alle Produkte sind wahlweise auf 5 1/4-Zoll- und 3 1/2-Zoll-Disketten erhältlich. Bitte beachten Sie die jeweiligen Bestellnummern. Bitte benutzen Sie die Bestellkarte.

Soweit nicht anders vermerkt, sind alle Produkte auf MS-DOS-Computern ab DOS 2.1, minimal 256 kByte RAM und den Grafikkarten CGA, MGA, EGA, VGA und Hercules lauffähig.

Legende:
(M) = mit universell verwendbarem Maustreiber, (D) = Demo-Diskette für DM 5,– erhältlich

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

```
[6F5]  259: 2C 00 2E A3 19 01 8C C8 87 06 2E 00 2E A3 1B 01
[TW5]  260: E4 21 24 F7 E6 21 1F EC 8A C1 EB 1F 8A 47 FE 3C
[4D5]  261: 01 74 12 3C 03 75 02 B0 01 B4 3D CD 21 73 0C 80
[8R5]  262: 7F FE 00 74 64 B1 20 B4 3C CD 21 8A 67 FE 8B 5F
[TK5]  263: FC 1F 88 07 C6 47 04 00 C6 47 10 1A 3C FD 73 46
[NM5]  264: 80 FC 03 75 0D 8A D8 33 C9 33 D2 B8 02 42 CD 21
[VR5]  265: EB 2C 80 FC 00 75 0B C6 47 10 00 BA 39 03 FF D2
[DN5]  266: EB 1C 80 FC 02 75 17 53 8A 1F 33 C9 33 D2 B8 02
[VK5]  267: 42 CD 21 5B 72 08 0B C2 74 04 C6 47 10 00 8B EC
[4L5]  268: 8A 46 02 88 47 1A C2 06 01 8B 5F FC 1F C6 07 00
[WG5]  269: C6 47 04 00 C6 47 10 1A C2 06 01 8B EC 8B 5E 02
[TS5]  270: BA 79 08 FF D2 C2 02 00 8B EC 8B 5E 02 80 3F FF
[7K5]  271: 75 14 B4 03 CD 10 A0 0B 00 2C 17 3A D0 72 0A B0
[1M5]  272: 0D BA A3 01 FF E2 8A 57 04 8A C2 B4 00 B9 0E 00
[8U5]  273: F6 F1 2A CC B0 20 51 BA A3 01 FF D2 59 E2 F5 C3
[FL5]  274: 58 8B EC 8E 46 00 89 46 00 BF 01 00 8B 5E 02 C6
[PK5]  275: 06 0E 00 00 57 BA 39 03 FF D2 5F 3C 00 74 F5 3C
[UN5]  276: 1A 74 1F 3C 0D 74 09 81 FF 00 01 73 E7 AA EB E4
[AN5]  277: 80 3F FF 74 0D 80 7F 10 0A 75 07 57 BA 39 03 FF
[CP5]  278: D2 5F 8D 45 FF 26 A2 00 00 BA 39 05 FF E2 33 DB
[EU5]  279: B0 3F BA A3 01 FF D2 B0 20 BA A3 01 FF E2 81 EC
[AX5]  280: FE 00 8B EC FF B6 FE 00 8C D0 8E C0 8D 7E 01 8B
[8F5]  281: 9E 02 01 80 26 0E 00 7F E8 54 00 74 49 3C 20 74
[VE5]  282: F7 3C 22 75 21 E8 47 00 74 3C 3C 22 74 0D 57 2B
[VT5]  283: FD 81 FF 00 01 5F 73 ED AA EB EA E8 31 00 74 26
[XR5]  284: 3C 2C 75 F7 EB 20 3C 2C 74 10 57 2B FD 81 FF 00
[6H5]  285: 01 5F 73 01 AA E8 17 00 75 EC 4F 3B FD 74 06 26
[UM5]  286: 80 3D 20 74 F5 47 8D 45 FF 2B C5 88 46 00 C3 57
[FR5]  287: BA 39 03 FF D2 5F 3C 1A 74 15 3C 0D 75 11 80 7F
[1L5]  288: 10 0A 75 09 57 BA 39 03 FF D2 5F B0 0D 3C 0D C3
[BK5]  289: 8B EC 8A 46 02 B4 00 89 86 00 01 C2 FE 00 BB 04
[HR5]  290: 00 BA 79 08 FF D2 4B 75 F8 C3 8B EC 8B 5E 02 B0
[AG5]  291: FF 38 07 74 06 80 7F 10 1A 74 02 F6 D0 98 89 46
[2T5]  292: 02 C3 81 EC FE 00 8B EC B4 0B CD 21 B4 FF 3C 00
[KH5]  293: 74 08 B4 08 CD 21 8A E0 B0 01 89 46 00 80 FC 00
[HM5]  294: 75 0A B4 08 CD 21 88 46 02 FE 46 00 FF A6 FE 00
[LP5]  295: BA 54 05 FF D2 33 D2 A8 01 74 01 4A 89 96 FE 01
[DM5]  296: C2 FE 01 BA 54 05 FF D2 33 D2 A8 02 74 01 4A 89
[VM5]  297: 96 FE 01 C2 FE 01 BA 54 05 FF D2 33 D2 A8 04 74
[KR5]  298: 01 4A 89 96 FE 01 C2 FE 01 BA 54 05 FF D2 33 D2
[7R5]  299: A8 03 74 01 4A 89 96 FE 01 C2 FE 01 BA 54 05 FF
[3P5]  300: D2 33 D2 A8 05 74 01 4A 89 96 FE 01 C2 FE 01 BA
[BK5]  301: 54 05 FF D2 33 D2 A8 01 75 01 4A 89 96 FE 01 C2
[AR5]  302: FE 01 5E 5B 59 81 EC 00 01 8B EC 88 4E 00 E3 16
[5N5]  303: C6 06 0E 00 FF 45 51 BA 39 03 FF D2 59 3C 1A 74
[YU5]  304: 05 88 46 00 E2 EF 8B EC 28 4E 00 FF E6 8B EC 8B
[6R5]  305: 5E 02 BA 39 05 FF E2 8B EC 8A 8E 02 01 8A 6E 02
[ZN5]  306: 80 FD 00 74 2B 8B 9E 02 02 8A C3 FE C8 02 C5 72
[NS5]  307: 1F 3A C1 77 1B 8B F3 BF 01 00 8A D5 46 47 8A 82
[LR5]  308: 01 01 3A 43 01 75 06 FE CA 75 F1 EB 05 43 EB D9
[EU5]  309: 33 DB 89 9E 02 02 C2 00 02 8B F4 8C DA 8C D0 8E
[H16]  310: D8 AD AD 8E C0 33 FF FC AC AA 8A C8 B5 00 F3 A4
[MM5]  311: 8E DA C2 02 01 8B EC 8B 46 02 87 46 04 83 C4 04
[8Z5]  312: FF E0 8B F4 83 C6 02 8C DA 8C D0 8E D8 8E C0 8B
[6L5]  313: 9C 00 01 8A 0C B5 00 03 F1 8D 7C 02 41 FD F3 A4
[NV5]  314: FC 8E DA 83 C4 04 FF E3 8B EC 8E 06 19 00 8B 7E
[AH5]  315: 04 8A 46 02 AA C2 04 00 8B EC 8E 06 19 00 8B 7E
[XG5]  316: 02 26 8A 05 B4 00 89 46 02 C3 8B EC 8B 4E 02 33
[SG5]  317: C0 80 F9 01 72 02 8A C5 89 86 00 01 C2 FE 00 8B
[WH5]  318: EC 8A 46 04 8A 66 02 3A E0 73 02 8A C4 88 46 04
[6R5]  319: C2 02 00 8B FC 83 C7 04 36 8A 4D FE 36 8A 1D 2A
[KN5]  320: D9 76 16 B7 00 8D 71 01 8C DA 8C D0 8E D8 8E C0
[MS5]  321: B5 00 8A C1 AA F3 A4 8E DA C2 02 00 8B FC 83 C7
[4W5]  322: 06 8C DA 8C D0 8E D8 8E C0 8A 5D FE 80 FB 00 75
[RP5]  323: 02 B3 01 B7 00 8D 31 8A 0D B5 00 FE CB 2A CB 73
[8P5]  324: 02 B1 00 8A 45 FC 3A C8 76 02 8A C8 8A C1 AA F3
[MS5]  325: A4 8E DA C2 04 00 8B FC 83 C7 04 8C D0 8E C0 26
[1K5]  326: 8A 8D FE 00 B5 00 26 8B 45 FE 3C 00 75 02 B1 00
[PU5]  327: 8A C1 AA 8A C4 F3 AA C2 02 00 8B EC 8B 46 04 8E
[HL5]  328: C0 8C DB 2B C3 B1 04 D3 E0 40 8B 5E 02 89 47 01
[VF5]  329: 26 A0 00 00 88 07 89 5E 04 C2 02 00 8B EC 8C C8
[PT5]  330: 87 46 02 8C DB 8E C3 FF 36 19 00 50 CB 8B F4 83
[9S5]  331: C6 02 8C DA 8C D0 8E D8 8A 8C 00 01 B5 00 AC 3A
[RD5]  332: C1 73 02 8A C8 8E 84 03 01 8A 9C 01 01 B7 00 80
[VL5]  333: FB 00 77 01 43 8B FB 26 A0 00 00 2A C3 72 0A FE
[0U5]  334: C0 3A C8 76 02 8A C8 F3 A4 8E DA C2 06 01 81 EC
[9U5]  335: FE 00 8B EC B4 2C CD 21 C6 46 00 08 8A C5 E8 1F
[FD5]  336: 00 89 46 01 C6 46 03 3A 8A C1 E8 13 00 89 46 04
[TM5]  337: C6 46 06 3A 8A C6 E8 07 00 89 46 07 FF A6 FE 00
[SM5]  338: B4 00 B3 0A F6 F3 05 30 30 C3 8B EC 8A 46 02 8B
[2T5]  339: 76 04 8B 5E 06 88 04 D1 E3 89 77 1C AC 8A D0 AD
[5M5]  340: B4 00 8B C8 AD 8E C0 26 88 0E 00 00 BF 01 00 B0
[WV5]  341: 20 F3 AA FE CA 75 E8 C2 06 00 8B EC 8B 5E 06 D1
[5M5]  342: E3 8B 77 1C D1 E8 8A 47 1A B4 00 50 8B 56 02 8B
[0V5]  343: 4E 04 80 C9 80 80 FD 98 73 08 D0 E9 D1 DA FE C5
[LS5]  344: EB F3 83 EA 01 80 D9 00 F6 E1 8B C8 58 50 F7 E2
[NM5]  345: 03 D1 8B D0 8B FB 8A 1F B8 00 42 CD 21 59 57 03
[XU5]  346: C1 83 D2 00 8E C0 8B EA AC B4 00 8B F8 AD 8B C8
[3P5]  347: 1E AD 8E D8 88 0E 00 00 BA 01 00 B4 3F CD 21 1F
[PE5]  348: 72 24 3B C1 75 20 4F 75 E4 33 C9 33 D2 B8 02 42
[4M5]  349: CD 21 72 12 3B EA 75 06 8C C6 3B F0 74 08 5B C6
[KE5]  350: 47 10 00 C2 06 00 5B C6 47 10 1A C2 06 00 8B F4
[VM5]  351: 83 C6 02 8C DA 8C D0 8E D8 8E 84 00 01 BF 01 00
[SE5]  352: 26 8A 0E 00 00 B5 00 B3 00 AC 3A C1 73 06 8A D9
[SW5]  353: 2A D8 8A C8 F3 A4 8A CB B0 20 F3 AA 8E DA C2 02
[FP5]  354: 01 8B F4 83 C6 02 8C DA 8C D0 8E D8 8E 84 00 01
[0K5]  355: BF 01 00 26 8A 1E 00 00 B5 00 B1 00 AC 3A C3 73
[RX5]  356: 06 8A CB 2A C8 8A D8 B0 20 F3 AA 8A CB F3 A4 8E
[FV5]  357: DA C2 02 01 8B EC 8B 5E 06 D1 E3 8B 77 1C D1 EB
[2E5]  358: 8A 47 1A B4 00 50 8B 56 02 8B 4E 04 80 C9 80 80
[5V5]  359: FD 98 73 08 D0 E9 D1 DA FE C5 EB F3 83 EA 01 80
[GU5]  360: D9 00 F6 E1 8B C8 58 F7 E2 03 D1 8B D0 8A 1F B8
[UT5]  361: 00 42 CD 21 AC B4 00 8B F8 AD 8B C8 1E AD 8E D8
[NG5]  362: BA 01 00 B4 40 CD 21 1F 72 07 3B C1 75 03 4F 75
[PG5]  363: E8 C2 06 00 8B EC 8B 7E 00 8B 46 02 B3 00 23 C0
[8N5]  364: 74 15 BB 90 7F 79 04 F7 D8 B7 FF 23 C0 78 06 D1
[WS5]  365: E0 FE CB EB F6 22 E7 C6 46 00 00 89 46 01 88 5E
[3S5]  366: 03 FF E7 81 EC FA 00 8B FC B2 00 36 8A B5 FC 00
[UT5]  367: 36 8B B5 FD 00 36 8A 0D FC 80 D0 8E C0 47 B0
[4K5]  368: 20 23 F6 79 02 B0 2D AA 80 FB 00 75 14 B0 30 AA
[1X5]  369: 8B EC 8D 45 FF 2B C5 88 46 00 B0 20 AA FF A6 FA
[1H5]  370: 00 81 CE 00 80 33 C9 80 FB 81 72 03 E9 D1 00 E8
[4M5]  371: 7A 00 FE C9 80 FB 80 72 F6 81 C2 80 00 83 D6 00
[YP5]  372: 73 06 D1 DE D1 DA FE C3 80 FB 81 73 05 E8 5C 00
[US5]  373: FE C9 E8 86 00 2A E9 80 FD 07 77 33 8B EC 8B 46
[YL5]  374: 02 86 C4 89 46 02 83 C5 03 3B FD 75 01 47 8B F7
[DK5]  375: F6 D1 B5 00 03 F9 57 4E 4F 36 8A 04 36 88 05 3B

[9T5]  376: F5 77 F4 E3 07 4F 36 C6 05 30 E2 F9 5F EB 81 B0
[ST5]  377: 45 AA B0 2B 22 C9 79 04 F6 D9 B0 2D AA 8A C1 B4
[2T5]  378: 00 B1 0A F6 F1 0D 30 30 AB E9 64 FF 8B C2 8B EE
[AW5]  379: D1 ED D1 D8 D1 ED D1 D8 13 D0 13 F5 73 06 D1 DE
[RR5]  380: D1 DA FE C3 80 C3 03 C3 B0 00 D1 E2 D1 D6 D0 D0
[0I6]  381: FE CB 80 FB 80 77 F3 0C 30 AA C3 E8 EA FF B0 2E
[00G]  382: AA E8 C8 FF E8 E1 FF FE C5 80 FD 06 72 F3 26 8A
[9X5]  383: 45 FF 3C 30 75 05 FE CD 4F EB F3 3C 2E 74 F9 C3
[4R5]  384: E8 4B 00 FE C1 80 FB 81 73 F6 FE C9 81 C2 80 00
[RX5]  385: 83 D6 00 73 06 D1 DE D1 DA FE C3 E8 8E FF E8 BA
[YW5]  386: FF 80 F9 07 72 03 E9 66 FF 8B EC 83 C5 03 FE C9
[TN5]  387: 78 0B 8B 46 00 86 C4 89 46 00 45 EB F1 3B FD 73
[PU5]  388: 02 8B FD 36 80 7D FF 2E 75 01 4F E9 C2 FE 52 BD
[3T5]  389: 0A 00 8B C6 33 D2 F7 F5 8B F0 58 F7 F5 92 B4 1A
[FX5]  390: F6 E4 D0 E0 D1 D2 D1 D6 FE CB 23 F6 79 F4 C3 8B
[YS5]  391: EC 80 76 04 80 C3 BA 86 05 FF E2 83 EC 02 8B EC
[LD5]  392: 8B 46 06 87 46 04 87 46 02 89 46 00 8B 46 08 B3
[WP5]  393: 00 23 C0 74 15 BB 90 7F 79 04 F7 D8 B7 FF 23 C0
[6S5]  394: 78 06 D1 E0 FE CB EB F6 22 E7 C6 46 06 00 89 46
[DV5]  395: 07 88 5E 09 C3 BA 22 06 FF E2 BA BE 06 FF E2 BA
[7S5]  396: 4A 07 FF E2 8B EC 8B 46 08 23 C0 74 2E 50 24 7F
[WX5]  397: 50 FF 76 06 BA 4A 07 FF D2 8B EC FF 76 0A FF 76
[6S5]  398: 08 BA 22 06 FF D2 BA D2 08 FF D2 58 5B 59 80 E1
[2R5]  399: 80 0A D9 8B EC 89 46 06 89 5E 08 C2 04 00 8B EC
[PP5]  400: 8B 5E 04 4B 78 15 D1 E3 03 5E 02 8B 56 00 83 EA
[3L5]  401: 03 3B DA 73 06 2E 8B 07 89 46 00 C2 04 00 8B EC
[GR5]  402: 8B 5E 04 4B 78 1B D1 E3 03 5E 02 8B 56 00 83 EA
[EH5]  403: 03 3B DA 73 0C 2E 8B 07 87 46 00 89 46 04 C2 02
[ER5]  404: 00 C2 04 00 BA E5 09 FF E2 8B EC 8B 46 03 8A 6E
[YP5]  405: 05 22 ED 74 16 8A DC 80 CC 80 B1 90 2A CD 72 0B
[PG5]  406: D3 E8 15 00 00 22 DB 79 02 F7 D8 89 46 04 C2 02
[6H5]  407: 00 07 BA 41 08 FF D2 33 C0 80 FA 01 75 01 48 89
[TL5]  408: 46 08 06 C2 06 00 07 BA 41 08 FF D2 33 C0 80 FA
[UH5]  409: FF 75 01 48 89 46 08 06 C2 06 00 07 BA 41 08 FF
[YC5]  410: D2 33 C0 80 FA 00 75 01 48 89 46 08 06 C2 06 00
[7L5]  411: 07 BA 41 08 FF D2 33 C0 80 FA FF 74 01 48 89 46
[XH5]  412: 08 06 C2 06 00 07 BA 41 08 FF D2 33 C0 80 FA 01
[1H5]  413: 74 01 48 89 46 08 06 C2 06 00 07 BA 41 08 FF D2
[RE5]  414: 33 C0 80 FA 00 74 01 48 89 46 08 06 C2 06 00 8B
[EM5]  415: EC 8B 46 02 87 46 04 87 46 06 83 C4 04 FF E0 8B
[2K5]  416: EC 33 D2 8B 46 04 80 FC 00 74 07 42 22 C0 79 02
[4L5]  417: F7 DA 89 56 04 C2 02 00 8B EC 8A 4E 02 22 C0 74
[RP5]  418: 02 FE C9 8B 5E 04 80 3F FF 74 25 38 4F 04 76 10
[MS5]  419: 51 B0 0D BA A3 01 FF D2 B0 0A BA A3 01 FF D2 59
[CN5]  420: 38 4F 04 73 42 51 B0 20 BA A3 01 FF D2 59 EB F0
[HP5]  421: B4 0F CD 10 2A CC 73 FC 02 CC 51 B4 03 CD 10 59
[HT5]  422: 51 3A D1 76 09 33 DB B0 0D BA A3 01 FF D2 B4 0F
[VK5]  423: CD 10 B4 03 CD 10 59 3A D1 73 0C 51 33 DB B0 20
[9U5]  424: BA A3 01 FF D2 EB E7 C2 02 00 8B EC 8B 4E 02 83
[HN5]  425: F9 00 76 0E 8B 5E 04 51 B0 20 BA A3 01 FF D2 59
[AP5]  426: E2 F5 C2 02 00 8B EC B4 0F CD 10 B4 03 CD 10 8A
[CR5]  427: D6 B6 00 42 87 56 00 FF E2 8B EC B4 0F CD 10 B4
[YL5]  428: 03 CD 10 B6 00 42 87 56 00 FF E2 8B EC 8A 4E 02
[AG5]  429: 8A 6E 04 89 0E 24 00 83 7E 06 00 75 03 B9 00 1F
[HP5]  430: B4 01 CD 10 C2 06 00 8B EC B4 0F CD 10 B4 03 CD
[YL5]  431: 10 81 E9 00 1F 87 4E 00 FF E1 8B EC A0 25 00 B4
[AF5]  432: 00 87 46 00 FF E0 8B EC A0 24 00 B4 00 87 46 00
[US5]  433: FF E0 8B EC A0 09 00 8A D8 25 0F 00 B1 04 D2 EB
[WK5]  434: 80 E3 10 0A C3 87 46 00 FF E0 8B EC A0 09 00 B1
[NN5]  435: 04 D2 E8 25 07 00 87 46 00 FF E0 1E 8C D0 8E D8
[GS5]  436: 8B F4 83 C6 04 AC 8A C8 B5 00 33 DB 33 D2 33 ED
[WP5]  437: E3 4D E8 60 01 74 48 3C 2B 74 F7 3C 2D 75 04 F7
[SN5]  438: D5 EB EF 3C 26 75 6A E8 4B 01 74 33 3C 68 74 04
[ZF5]  439: 3C 48 75 2B E8 3E 01 74 26 2C 30 72 22 3C 0A 72
[UE5]  440: 12 2C 07 3C 0A 72 18 3C 10 72 08 2C 20 72 10 3C
[AM5]  441: 10 73 0C D1 E2 D1 E2 D1 E2 D1 E2 02 D0 EB D5 23
[0M5]  442: D2 74 1C B3 90 79 04 F7 DA F7 D5 22 F6 78 06 D1
[5W5]  443: E2 FE CB EB F6 80 E6 7F 23 ED 74 03 80 CE 80 8B
[FL5]  444: EC 88 BE 00 01 89 96 01 01 88 9E 03 01 1F C2 FC
[BM5]  445: 00 55 33 FF 3C 2E 75 53 23 FF 75 06 4F E8 D5 00
[DG5]  446: 75 F2 23 FF 74 01 47 23 FF 74 2A 57 53 52 78 12
[XN5]  447: B8 20 84 50 33 C0 50 BA 22 06 FF D2 5A 5B 5F 4F
[LZ5]  448: EB E5 B8 4C 7D 50 B8 CD CC 50 BA 22 06 FF D2 5A
[3R5]  449: 5B 5F 47 EB D2 58 23 C0 74 03 80 F3 80 8B EC 89
[KH5]  450: 96 00 01 89 9E 02 01 1F C2 FC 00 3C 65 74 04 3C
[UG5]  451: 45 75 42 23 FF 74 01 47 33 ED 8E C5 E8 76 00 74
[8U5]  452: A6 3C 2B 74 F7 3C 2D 75 08 8C C0 F7 D0 8E C0 EB
[7M5]  453: EB 2C 30 72 10 3C 0A 73 0C 95 B4 0A F6 E4 03 E8
[RU5]  454: E8 52 00 75 EC 8B C5 98 8C C5 23 ED 79 02 F7 D8
[AN5]  455: 03 F8 E9 72 FF B4 00 2C 30 73 03 E9 64 FF 74 0E
[ZU5]  456: 3C 0A 73 F7 B4 88 FE CC D0 E0 79 FA 24 7F 51 56
[TC5]  457: 57 50 33 C0 50 B8 20 84 50 33 C0 50 53 52 BA 22
[HW5]  458: 06 FF D2 BA 86 05 FF D2 5A 5B 5F 5E 59 23 FF 74
[HR5]  459: 01 4F E9 28 FF 23 C9 74 05 AC 3C 20 E1 F7 C3 8B
[WP5]  460: EC 8B 46 05 8A 6E 07 22 ED 74 16 8A DC 80 CC 80
[KM5]  461: B1 90 2A CD 72 0B D3 E8 15 00 00 22 DB 79 02 F7
[WG5]  462: D8 89 46 06 8B 46 02 89 46 04 C2 02 00 8B EC 33
[VB5]  463: D2 80 7E 02 00 74 01 4A 89 96 00 01 C2 FE 00 81
[EN5]  464: EC FC 00 8B EC 8B 86 FE 00 89 46 01 C6 46 00 02
[2T5]  465: FF A6 FC 00 81 EC FA 00 8B EC 8B 86 FC 00 89 46
[LK5]  466: 01 8B 86 FE 00 89 46 03 C6 46 00 04 FF A6 FA 00
[2K5]  467: 8B EC 8B 46 04 22 C0 79 0A 8B 5E 02 89 1E 26 00
[CL5]  468: A3 28 00 FF 36 28 00 FF 36 26 00 22 E4 74 4B B8
[8N5]  469: A7 98 50 B8 E0 18 50 BA 22 06 FF D2 B8 28 68 50
[CN5]  470: B8 46 B1 50 BA 86 05 FF D2 B8 46 06 8B 5E 08 34
[KG5]  471: AA 80 F3 55 86 C3 80 F4 C3 B7 00 23 C0 75 04 22
[9V5]  472: DB 74 11 B7 80 22 DB 78 08 D1 E0 D0 D3 FE CF EB
[1R5]  473: F4 80 E3 7F 89 46 06 89 5E 08 8B EC 58 89 46 06
[HN5]  474: A3 26 00 58 89 46 08 A3 28 00 C3 8B EC B4 0F CD
[LL5]  475: 10 B4 03 CD 10 8B F2 8A 56 04 8A 76 06 81 EA 01
[AF5]  476: 01 B4 02 CD 10 B4 08 CD 10 8B 4E 02 23 C9 74 02
[7H5]  477: 8A C4 B4 00 89 46 06 8B D6 B4 02 CD 10 C2 04 00
[BT5]  478: 81 EC FE 00 8B EC B4 2A CD 21 8A C6 E8 38 00 89
[NK5]  479: 46 01 C6 46 03 2D 8A C2 E8 2C 00 89 46 04 C6 46
[0M5]  480: 06 2D B8 31 39 81 E9 6C 07 80 F9 64 72 06 B8 32
[0G5]  481: 30 80 E9 64 89 46 07 8A C1 E8 0B 00 89 46 09 C6
[PH5]  482: 46 00 0A FF A6 FE 00 B4 00 B3 0A F6 F3 05 30 30
[PV5]  483: C3 83 EC 02 B4 00 CD 1A 8A DA 8A D6 8A F1 8A CD
[BK5]  484: B5 00 22 DB 75 08 23 D2 75 04 23 C9 74 13 B5 A0
[0V5]  485: 22 C9 78 0A D0 E3 D1 D2 D0 D1 FE CD EB F2 80 E1
[5S5]  486: 7F 51 52 B8 60 7C 50 B8 7A F9 50 BA 22 06 FF D2
[XM5]  487: 8F 46 0A 58 87 46 0C FF E0 8B EC 33 C0 80 7E 02
[1L5]  488: 02 75 03 8B 46 03 89 86 00 01 C2 FE 00 8B EC 33
[XG5]  489: C0 33 DB 80 7E 02 04 75 06 8B 46 03 8B 5E 05 89
[MN5]  490: 86 FE 00 89 9E 00 01 C2 FC 00 8B EC 80 7E 05 98
[3S5]  491: 73 30 B8 FF FF B3 FF B1 90 2A 4E 05 73 07 80 C1
[DH5]  492: 08 D2 E0 EB 17 B0 00 80 E9 08 73 07 80 C1 08 D2
```

```
[CG5]  493: E4 EB Ø9 B4 ØØ 8Ø F9 Ø8 73 Ø9 D2 E3 21 46 Ø2 2Ø
[US5]  494: 5E Ø4 C3 89 46 Ø2 89 46 Ø4 C3 81 EC FC ØØ 8B EC
[LM5]  495: 8B 86 FE ØØ BB 1Ø ØØ 33 C9 33 D2 F7 F3 52 23 CØ
[HL5]  496: EØ F7 F7 D9 88 4E ØØ BF Ø1 ØØ 58 Ø4 3Ø 3C 39 76
[SN5]  497: Ø2 Ø4 Ø7 88 Ø3 47 E2 F2 FF A6 FC ØØ 8B EC 8B 56
[BK5]  498: Ø2 EC B4 ØØ 89 46 Ø2 C3 8B EC 83 C5 Ø2 8B 9E ØØ
[RC5]  499: Ø1 8A 4E ØØ B5 ØØ E3 11 45 8A 46 ØØ 3C 2Ø 74 Ø7
[EP5]  5ØØ: 51 BA A3 Ø1 FF D2 59 E2 EF C2 ØØ Ø1 8B EC 83 C5
[YL5]  5Ø1: Ø2 8B 9E ØØ Ø1 BØ 22 BA A3 Ø1 FF D2 8A 4E ØØ B5
[7M5]  5Ø2: ØØ E3 ØD 45 8A 46 ØØ 51 BA A3 Ø1 FF D2 59 E2 F3
[6R5]  5Ø3: BØ 22 BA A3 Ø1 FF D2 C2 ØØ Ø1 8B EC 8B 5E Ø2 BØ
[SU5]  5Ø4: 2C BA A3 Ø1 FF E2 8B EC 8A 5E Ø2 8A 4E Ø4 8B 56
[BM5]  5Ø5: Ø6 EC 32 C3 22 C1 74 F9 C2 Ø6 ØØ 8B EC 83 C5 Ø3
[7S5]  5Ø6: 8A 4E FF B5 ØØ E8 15 ØØ 8A FE E8 1Ø ØØ 8A DE E8
[4P5]  5Ø7: ØB ØØ B2 32 8B CB B4 2D CD 21 C2 ØØ Ø1 B6 ØØ E8
[LG5]  5Ø8: 14 ØØ 2C 3Ø 72 19 3C ØA 73 15 5Ø BØ ØA F6 E6 8A
[KH5]  5Ø9: FØ 58 Ø2 FØ EB E9 BØ ØØ E3 Ø5 8A 46 ØØ 45 49 C3
[DK5]  51Ø: 8B EC 8B 76 Ø2 8B 7E Ø4 8B Ø4 87 Ø5 89 Ø4 C2 Ø4
[MS5]  511: ØØ 8B EC 8C D8 8E C0 8B 76 Ø2 8B 7E Ø4 26 8B Ø5
[HS5]  512: A5 89 44 FE 26 8B Ø5 A5 89 44 FE C2 Ø4 ØØ 8B EC
[ZG5]  513: 8C DA 8E 5E Ø2 8E 46 Ø4 8A ØE ØØ ØØ 26 8A 2E ØØ
[AM5]  514: ØØ 3A CD 73 Ø2 8A CD B5 ØØ 41 33 F6 33 FF 26 8A
[CR5]  515: Ø5 A4 88 44 FF E2 F7 8E DA C2 Ø4 ØØ 8B EC B8 11
[LT5]  516: 95 5Ø B8 FØ A6 5Ø FF 76 Ø8 FF 76 Ø6 BA BE Ø6 FF
[8R5]  517: D2 58 5B 22 FF 74 12 8Ø CB 8Ø 8A C4 8A E3 8Ø FF
[ØS5]  518: 9Ø 73 Ø6 D1 E8 FE C7 EB F5 8B 5E ØE 22 FF 74 13
[5V5]  519: 8B 4E ØC 8Ø CB 8Ø 8Ø FF 9Ø 73 Ø8 DØ EB D1 D9 FE
[KR5]  52Ø: C7 EB F3 22 DB 75 Ø2 E3 2C 5Ø BØ B6 E6 43 BØ 14
[GM5]  521: E6 43 BØ ØØ E6 4Ø 58 E6 4Ø 8A C4 E6 42 E4 61 ØC
[CM5]  522: Ø3 E6 61 E4 4Ø 22 CØ 79 FA E4 4Ø 22 CØ 78 FA E2
[SR5]  523: F2 FE CB 79 EE E4 61 24 FD E6 61 BØ 36 E6 43 BØ
[1K5]  524: ØØ E6 4Ø E6 4Ø C2 Ø8 ØØ 81 EC FE ØØ 8B EC C6 46
[DT5]  525: ØØ ØØ FF A6 FE ØØ 8B ØE ØF ØØ B4 Ø1 CD 1Ø 83 EC
[UK5]  526: Ø2 8B EC 8B 76 Ø2 8A 46 Ø4 B4 ØD 89 46 Ø1 C6 46
[6A5]  527: Ø3 ØØ 22 CØ 74 1E Ø4 Ø3 B4 2Ø 89 46 Ø1 C7 46 Ø3
[FM5]  528: 2F 43 B4 ØØ 8B F8 81 FF ØØ Ø1 76 Ø3 BF ØØ Ø1 C7
[GE5]  529: 43 Ø2 ØD ØØ 1E 56 16 45 55 8E 1E 2A ØØ 8E 1E 2C
[AG5]  53Ø: ØØ 1E 33 F6 E8 Ø8 ØØ 43 4F 4D 53 5Ø 45 43 1D 5B
[5N5]  531: 8B D6 AC 22 CØ 74 1B 8B FB B9 Ø8 ØØ 24 DF 47 2E
[9W5]  532: 3A 45 FF AC E1 F6 74 Ø7 AC 22 CØ 75 FB EB E1 83
[5R5]  533: C2 Ø8 8C DØ 8E CØ 8B DC E8 ØØ ØØ 5F 2E 8C 55 12
[DM5]  534: 9Ø 2E 89 65 17 9Ø B8 ØØ 4B CD 21 FA B8 ØØ ØØ 8E
[KU5]  535: DØ BC ØØ ØØ FB 8B EC 8E 5E Ø8 8B 6E Ø6 81 C4 ØE
[AN5]  536: Ø1 B4 ØF CD 1Ø B4 Ø3 CD 1Ø 8Ø FE 18 72 ØB B8 ØA
[5L5]  537: ØE CD 1Ø B6 17 B4 Ø2 CD 1Ø 8B ØE 24 ØØ B4 Ø1 CD
[BR5]  538: 1Ø FF E5 8B EC B4 ØF CD 1Ø B4 Ø3 CD 1Ø 8A 46 Ø2
[FL5]  539: B4 Ø5 CD 1Ø B4 ØF CD 1Ø B4 Ø2 CD 1Ø C2 Ø6 ØØ 8C
[XS5]  54Ø: DA 83 C2 Ø3 83 EB Ø3 74 ØF 33 CØ 8E C2 33 FF B9
[UT5]  541: Ø8 ØØ F3 AB 42 4B 75 F3 BB Ø4 ØØ BA 79 Ø8 FF D2
[FY5]  542: 4B 75 F8 C3 1E 8C DØ 8E D8 8B FC 83 C7 Ø5 8A 5D
[SY5]  543: FF 8Ø FB FF 75 Ø2 FE CB B7 ØØ 88 39 8B D7 B4 39
[CS5]  544: CD 21 1F C2 ØØ Ø1 1E 8C DØ 8E D8 8B FC 83 C7 Ø5
[VU5]  545: 8A 5D FF 8Ø FB FF 75 Ø2 FE CB B7 ØØ 88 39 8B D7
[4T5]  546: B4 3A CD 21 1F C2 ØØ Ø1 1E 8C DØ 8E D8 8B FC 83
[CY5]  547: C7 Ø5 8A 5D FF 8Ø FB FF 75 Ø2 FE CB B7 ØØ 88 39
[DP5]  548: 8B D7 B4 3B CD 21 1F C2 ØØ Ø1 5A 8F Ø6 26 ØØ 8F
[SS5]  549: Ø6 28 ØØ FF E2 B8 Ø7 ØE CD 1Ø C3 81 EC FC ØØ 8B
[LU5]  55Ø: FC 8C DØ 8E CØ 36 8A 85 FE ØØ B4 ØØ 8B C8 36 8B
[SU5]  551: 95 FC ØØ AA BØ 2Ø F3 AA FF E2 8B EC 8E Ø6 19 ØØ
[9T5]  552: 1E 8C DØ 8E D8 8D 76 Ø6 8A ØC B5 ØØ 8Ø F9 FA 72
[LE5]  553: Ø2 B1 FA Ø3 F1 8D 7C Ø1 8Ø F9 Ø4 76 Ø2 B1 Ø4 E3
[VH5]  554: Ø8 8Ø 3C 2E 74 ØF 4E E2 F8 C7 Ø5 2E 42 C7 45 Ø2
[2G5]  555: 41 53 83 C7 Ø4 C6 Ø5 ØØ 8D 56 Ø7 B1 2Ø B4 3C CD
[AM5]  556: 21 72 33 8B D8 8C C0 87 46 Ø4 89 46 Ø6 B4 FD 87
[XC5]  557: 46 Ø2 89 46 Ø8 B9 Ø7 ØØ 8D 56 Ø3 B4 4Ø CD 21 72
[NM5]  558: 11 3B C1 75 ØD 8E 5E Ø4 8B 56 Ø6 8B 4E Ø8 B4 4Ø
[3R5]  559: CD 21 B4 3E CD 21 1F C2 Ø4 Ø1 8B EC 8E Ø6 19 ØØ
[1T5]  56Ø: 1E 8C DØ 8E D8 8D 76 Ø4 8A ØC B5 ØØ 8Ø F9 FA 72
[LE5]  561: Ø2 B1 FA Ø3 F1 8D 7C Ø1 8Ø F9 Ø4 76 Ø2 B1 Ø4 E3
[VH5]  562: Ø8 8Ø 3C 2E 74 ØF 4E E2 F8 C7 Ø5 2E 42 C7 45 Ø2
[BF5]  563: 41 53 83 C7 Ø4 C6 Ø5 ØØ 8D 56 Ø5 B8 ØØ 3D CD 21
[LG5]  564: 72 35 8B D8 B9 Ø7 ØØ B4 3F CD 21 72 26 3D Ø7 ØØ
[2N5]  565: 75 21 8Ø 7E Ø5 FD 75 1B 8B 56 Ø2 83 FA FF 75 Ø8
[YS5]  566: 8E 5E Ø6 8B 56 Ø8 EB Ø4 8C CØ 8E D8 8B 4E ØA B4
[5R5]  567: 3F CD 21 B4 3E CD 21 1F C2 Ø2 Ø1 8B EC 8A 5E Ø2
[KP5]  568: B7 ØØ B4 ØB CD 1Ø C2 Ø2 ØØ 8B EC 8B 56 Ø4 8A 46
[PH5]  569: Ø2 EE C2 Ø4 ØØ BØ B6 E6 43 BØ 14 E6 43 BØ ØØ E6
[EL5]  57Ø: 4Ø 8B EC 8A 4E Ø2 83 C5 Ø3 E8 48 Ø2 22 CØ 75 ØD
[RF5]  571: BØ 36 E6 43 BØ ØØ E6 4Ø E6 4Ø C2 ØØ Ø1 3C 41 72
[WE5]  572: 2E 3C 47 77 2A 2C 41 B4 ØØ E8 Ø7 ØØ ØA ØC Ø1 Ø3
[KK5]  573: Ø5 Ø6 Ø8 5B Ø3 D8 2E 8A Ø7 Ø2 Ø6 2C ØØ 8A D8 E8
[DL5]  574: 12 Ø2 8A 3E 2D ØØ 3C 23 75 3Ø FE C3 E9 87 ØØ 3C
[MM5]  575: 5Ø 75 Ø4 BØ ØØ EB E6 3C 4F 75 11 E8 F6 Ø1 E8 Ø7
[NN5]  576: Ø2 5Ø BØ ØC F6 E2 A2 2C ØØ 58 EB 3C 4E 75 12
[6U5]  577: E8 E1 Ø1 E8 F2 Ø1 8A DA EB C8 3C 2B 75 52 FE C3
[6P5]  578: EB 54 3C 4C 75 ØD E8 CB Ø1 88 16 DC Ø1 88 16 2D ØØ
[GU5]  579: E9 79 FF 3C 54 75 ØD E8 BA Ø1 E8 CB Ø1 88 16 2E
[EG5]  58Ø: ØØ E9 68 FF 3C 3C 75 12 8Ø 3E 2C ØØ ØC 72 Ø5 8Ø
[3R5]  581: 2E 2C ØØ ØC E8 9D Ø1 E9 52 FF 3C 3E 75 35 8Ø 3E
[YK5]  582: 2C ØØ 48 73 Ø5 8Ø Ø6 2C ØØ ØC E8 87 Ø1 E9 3C FF
[SN5]  583: 3C 2D 75 Ø5 FE CB E8 7B Ø1 3C 3Ø 72 Ø9 3C 39 77
[4N5]  584: Ø5 E8 84 Ø1 8A FA B4 ØØ 3C 2E 75 33 FE C4 E8 63
[ZP5]  585: Ø1 EB F5 3C 4D 75 22 E8 5A Ø1 3C 4E 75 Ø7 C6 Ø6
[9L5]  586: 2F ØØ Ø7 EB 14 3C 4C 75 Ø7 C6 Ø6 2F ØØ Ø8 EB Ø9
[YK5]  587: 3C 53 75 Ø5 C6 Ø6 2F ØØ Ø6 E8 38 Ø1 E9 ED FE E8
[EK5]  588: AA ØØ Ø1 ØØ 43 47 43 43 7C 3F EC 3B 8F 38 62 35
[ZN5]  589: 63 32 8F 2F E4 2C 5F 2A FE 27 CØ 25 A1 23 A1 21
[EP5]  59Ø: BE 1F F6 1D 48 1C B1 1A 32 19 C8 17 72 16 3Ø 15
[HT5]  591: FF 13 EØ 12 D1 11 D1 1Ø DF ØF FB ØE 24 ØE 59 ØD
[WL5]  592: 99 ØC E4 ØB 39 ØB 98 ØA ØØ ØA 7Ø Ø9 E8 Ø8 68 Ø8
[PH5]  593: FØ Ø7 7E Ø7 12 Ø7 AC Ø6 4C Ø6 F2 Ø5 9D Ø5 4C Ø5
[RC5]  594: ØØ Ø5 B8 Ø4 74 Ø4 34 Ø4 F8 Ø3 BF Ø3 89 Ø3 56 Ø3
[XD5]  595: 26 Ø3 F9 Ø2 CE Ø2 A6 Ø2 8Ø Ø2 5C Ø2 3A Ø2 1A Ø2
[DE5]  596: FC Ø1 DF Ø1 C4 Ø1 AB Ø1 93 Ø1 7C Ø1 67 Ø1 53 Ø1
[CD5]  597: 4Ø Ø1 2E Ø1 1D Ø1 ØD Ø1 FE ØØ F9 ØØ E2 ØØ D6 ØØ
[4H5]  598: CA ØØ BE ØØ B4 ØØ A9 ØØ AØ ØØ 97 ØØ 5E 5Ø 8A C3
[SM5]  599: B4 ØØ D1 EØ Ø3 FØ 2E 8B Ø4 E6 42 8A C4 E6 42 8A
[3T5]  6ØØ: DF B7 ØØ D1 E3 D1 E3 D1 E3 D1 E3 58 5Ø 8A C4 B4
[CR5]  6Ø1: ØØ 8B F8 BØ AØ F6 26 2F ØØ 8B F7 4E 78 ØB BA Ø3
[8T5]  6Ø2: ØØ F7 E2 D1 EA D1 D8 EB F2 8A 16 2E ØØ B6 ØØ F7
[DR5]  6Ø3: E2 F7 F3 8B FØ B8 ØØ Ø5 4F 78 ØB BA Ø3 ØØ F7 E2
[DU5]  6Ø4: D1 EA D1 D8 EB F2 8A 16 2E ØØ B6 ØØ F7 E2 F7 F3
[8N5]  6Ø5: 8B F8 E4 61 ØC Ø3 E6 61 E4 4Ø 22 CØ 78 FA E4 4Ø
[GR5]  6Ø6: 22 CØ 79 FA 4E 75 Ø6 E4 61 24 FD E6 61 4F 75 E8
[VR5]  6Ø7: 58 E9 B8 FD BØ ØØ E3 ØF 8A 46 ØØ 3C 61 72 Ø6 3C
```

```
[ØN5]  6Ø8: 7A 77 Ø2 24 DF 45 49 C3 BA ØØ ØA 2C 3Ø 86 C2 F6
[MS5]  6Ø9: E6 Ø2 DØ E8 DE FF 3C 3Ø 72 Ø4 3C 39 76 ED C3 8B
[6N5]  61Ø: EC F6 46 Ø3 8Ø 79 Ø3 F7 5E Ø2 C3 8B EC 8Ø 66 Ø4
[8X5]  611: 7F C3 1E 8C DØ 8E D8 8B FC 83 C7 Ø5 8A 5D FF 8Ø
[5S5]  612: FB FF 75 Ø2 FE CB B7 ØØ 88 39 8B D7 B4 41 CD 21
[VS5]  613: 1F C2 ØØ Ø1 8B EC 83 C5 Ø3 8A 5E FF B7 ØØ 8B FB
[7R5]  614: 23 DB 74 ØC 8Ø 7B FF 3A 74 Ø6 8Ø 7B FF 5C 75 1Ø
[1M5]  615: 8Ø FB FC 77 ØB C7 Ø3 2A 2E C6 43 Ø2 2A 8Ø C3 Ø3
[KS5]  616: 8Ø FB FF 75 Ø2 B3 FE 88 5E FF Ø3 DD B2 ØØ 36 88
[SG5]  617: 17 8Ø 7E Ø1 3A 75 11 8A 46 ØØ 24 DF 2C 4Ø 72 Ø8
[4S5]  618: 3C 1A 77 Ø4 8A DØ EB Ø8 B4 19 CD 21 8A DØ FE C2
[NT5]  619: 52 33 DB 8A C2 Ø4 4Ø BA A3 Ø1 FF D2 BØ 3A BA A3
[ØN5]  62Ø: Ø1 FF D2 BØ 5C BA A3 Ø1 FF D2 5A 52 81 EC 81 ØØ
[5R5]  621: 8B F4 36 C6 Ø4 5C 46 1E 8C DØ 8E D8 B4 47 CD 21
[BR5]  622: 8A 4E FF B5 ØØ 8B FD Ø3 F9 4F 36 8Ø 3D 5C 74 Ø4
[4P5]  623: E2 F7 EB 27 3B FD 75 Ø1 47 36 FF 35 52 8B D5 36
[MK5]  624: C6 Ø5 ØØ B4 3B CD 21 5A 36 8F Ø5 72 ØE 83 C6 4Ø
[EP5]  625: B4 47 CD 21 8D 54 BF B4 3B CD 21 1F 36 8A Ø4 46
[GM5]  626: 22 CØ 74 Ø7 BA A3 Ø1 FF D2 EB F1 81 C4 81 ØØ BØ
[TW5]  627: ØD BA A3 Ø1 FF D2 B4 2F CD 21 8D 77 1E 33 DB 1E
[PR5]  628: 8C DØ 8E D8 8B D4 83 C2 Ø7 B1 31 B4 4E CD 21 1F
[WM5]  629: 73 Ø3 E9 93 ØØ 33 ED 8B FE 26 8A Ø3 22 CØ 74 1E
[AY5]  63Ø: 3C 2E 74 Ø8 BA A3 Ø1 FF D2 45 EB ED BØ 2E 83 FD
[AV5]  631: Ø8 73 F1 BØ 2Ø BA A3 Ø1 FF D2 45 4F EB EE 83 FD
[1M5]  632: ØC 73 ØA BØ 2Ø BA A3 Ø1 FF D2 45 EB F1 BD Ø6 ØØ
[7M5]  633: 26 F6 44 F7 1Ø 74 26 BØ 3C BA A3 Ø1 FF D2 BØ 44
[ØV5]  634: BA A3 Ø1 FF D2 BØ 49 BA A3 Ø1 FF D2 BØ 52 BA A3
[BR5]  635: Ø1 FF D2 BØ 3E BA A3 Ø1 FF D2 BD Ø1 ØØ BØ 2Ø BA
[WP5]  636: A3 Ø1 FF D2 4D 75 F6 B4 ØF CD 1Ø B4 Ø2 CD 1Ø 33
[UN5]  637: DB 8Ø C2 12 3A 16 ØB ØØ 72 Ø7 BØ ØD BA A3 Ø1 FF
[RS5]  638: D2 B4 4F CD 21 E9 68 FF B4 ØF CD 1Ø B4 Ø3 CD 1Ø
[ØS5]  639: 33 DB 22 D2 74 Ø7 BØ ØD BA A3 Ø1 FF D2 BØ 2Ø BA
[9S5]  64Ø: A3 Ø1 FF D2 5A B4 36 CD 21 F7 E1 F7 E3 BB ØA ØØ
[RM5]  641: 33 C9 8B FØ 8B C2 33 D2 F7 F3 96 F7 F3 52 96 23
[BT5]  642: F6 EØ F3 23 CØ 75 EF 33 DB F7 D9 58 ØC 3Ø 51 BA
[CM5]  643: A3 Ø1 FF D2 59 E2 F4 E8 ØF ØØ 2Ø 42 79 74 65 73
[6P5]  644: 2Ø 66 72 65 65 2Ø ØD ØD ØØ 5E 2E 8A Ø4 22 CØ 74
[FS5]  645: Ø8 BA A3 Ø1 FF D2 46 EB F1 C2 ØØ Ø1 1E 8C DØ 8E
[NV5]  646: D8 8E CØ 8B F4 81 C6 Ø5 Ø1 8A 5C FF 8Ø FB FF 75
[1W5]  647: Ø2 FE CB B7 ØØ 88 38 8B D6 8D BC ØØ FF 8A 5D FF
[XR5]  648: 8Ø FB FF 75 Ø2 FE CB 88 39 B4 56 CD 21 1F C2 ØØ
[HR5]  649: Ø2 8B EC 8B 5E Ø2 23 DB 75 74 53 33 DB 33 ED 33
[ZP5]  65Ø: F6 33 FF B9 7F ØØ E4 61 24 FC ØC Ø1 E6 61 BØ A4
[FN5]  651: E6 43 BØ ØØ E6 42 BØ 94 E6 43 BØ 1F E6 42 BA Ø1
[2N5]  652: Ø2 EE EC DØ E8 83 D3 ØØ DØ E8 83 D5 ØØ DØ E8 83
[SH5]  653: D6 ØØ DØ E8 83 D7 ØØ E4 42 24 1Ø 74 FA E4 42 24
[4W5]  654: 1Ø 75 FA E2 DD FB B1 Ø2 D3 E3 D3 E5 D3 E6 D3 E7
[3A5]  655: E8 Ø8 ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 8B C3 5B 2E 89
[5N5]  656: Ø7 2E 89 6F Ø2 2E 89 77 Ø4 2E 89 7F Ø6 5B E8 ØØ
[8R5]  657: ØØ 5E 83 E3 Ø3 D1 E3 2E 8B 4Ø E2 8B EC 89 46 Ø2
[TL5]  658: C3 8B EC BA Ø1 Ø2 EC 33 D2 8A 66 Ø2 8Ø E4 Ø6 75
[KP5]  659: Ø4 A8 1Ø EB 14 8Ø FC Ø4 75 Ø4 A8 2Ø EB ØB 8Ø FC
[TE5]  66Ø: Ø2 75 Ø4 A8 4Ø EB Ø2 A8 8Ø 75 Ø1 4A 89 56 Ø2 C3
[WS5]  661: BA D2 Ø8 FF E2 8B EC B2 ØØ 8A 76 Ø2 8B 7E Ø3 8A
[YL5]  662: 5E Ø5 81 CF ØØ 8Ø 8Ø FB ØØ 75 Ø1 C3 8Ø F3 8Ø ØA
[HZ5]  663: DB 79 ØE F6 DB DØ EB 73 Ø4 D1 EF D1 DA F6 DB EB
[AU5]  664: ØA DØ EB 73 Ø6 D1 EF D1 DA FE C3 8Ø F3 8Ø 88 5E
[ØM5]  665: Ø5 1E B9 18 ØØ BE Ø1 ØØ 33 CØ 8B E8 8B D8 D1 E2
[5X5]  666: D1 D7 D1 D5 D1 E2 D1 D7 D1 D5 2B EE 1B D8 D1 E2
[6W5]  667: D1 D7 D1 D5 D1 D3 D1 E2 D1 D7 D1 D5 D1 D3 D1 E6
[AX5]  668: D1 DØ 8E DB 8E C5 2B EE 1B D8 72 ØE 2B EE 1B D8
[VP5]  669: 72 Ø8 83 ED Ø1 83 DB ØØ 73 Ø6 8C DB 8C C5 EB Ø3
[CM5]  67Ø: 83 CE Ø1 E2 C9 83 C6 Ø1 15 ØØ ØØ D1 E8 D1 DE 1F
[YK5]  671: 8B EC 8A 5E Ø4 24 7F 8Ø E3 8Ø ØA C3 88 46 Ø4 89
[8R5]  672: 76 Ø2 C3 8B EC 83 C5 Ø3 8A 4E FF B5 ØØ E8 1E ØØ
[4P5]  673: 8A FA E8 19 ØØ 8A DA E8 14 ØØ 83 FA 64 73 Ø4 81
[HP5]  674: C2 6C Ø7 8B CA 8B D3 B4 2B CD 21 C2 ØØ Ø1 33 D2
[BH5]  675: E8 17 ØØ 2C 3Ø 72 1C 3C ØA 73 18 B4 ØØ 5Ø B8 ØA
[HM5]  676: ØØ F7 E2 8B DØ 58 Ø3 DØ EB E6 BØ ØØ E3 Ø5 8A 46
[UR5]  677: ØØ 45 49 C3 81 EC FC ØØ 8B EC 8C DØ 8E CØ 1E BE
[PM5]  678: 8Ø ØØ 8E 1E 2A ØØ AC 8A C8 B5 ØØ 8B 96 FE ØØ 23
[CM5]  679: D2 74 32 8D 7E Ø1 E8 34 ØØ 74 22 3C 2Ø 74 F7 3C
[RM5]  68Ø: 2C 74 1A 3C 61 72 Ø6 3C 7A 77 Ø2 2C 2Ø AA E8 1C
[CL5]  681: ØØ 74 ØA 3C 2Ø 74 Ø6 3C 2F 75 E4 4E 41 4A 75 D3
[FT5]  682: 8D 55 FF 2B D5 88 56 ØØ 1F FF A6 FC ØØ BØ ØD E3
[EX5]  683: Ø2 49 AC 3C ØD C3 BA 54 ØA FF E2 BA 47 ØA FF E2
[YY5]  684: 8B EC FF 76 Ø4 FF 76 Ø2 BA 54 ØA FF D2 8B EC FF
[VX5]  685: 76 Ø8 FF 76 Ø6 BA 47 ØA FF D2 BA BE Ø6 FF D2 8B
[BM5]  686: EC 8F 46 Ø6 8F 46 Ø8 C3 8B EC 8B 46 Ø4 24 7F 8Ø
[SL5]  687: FC 81 72 12 BB ØØ 81 53 33 DB 53 5Ø FF 76 Ø2 BA
[NØ6]  688: BE Ø6 FF D2 EB Ø4 5Ø FF 76 Ø2 8B EC FF 76 Ø2 FF
[AP5]  689: 76 ØØ FF 76 Ø2 FF 76 ØØ BA 22 Ø6 FF D2 B8 B3 76
[CR5]  69Ø: 5Ø B8 BD 83 5Ø 8B EC FF 76 Ø6 FF 76 Ø4 BA 22 Ø6
[2W5]  691: FF D2 B8 1E 79 5Ø B8 A6 F4 5Ø BA 86 Ø5 FF D2 8B
[DS5]  692: EC FF 76 Ø6 FF 76 Ø4 BA 22 Ø6 FF D2 B8 83 7B 5Ø
[MY5]  693: B8 BØ FC 5Ø BA 86 Ø5 FF D2 8B EC FF 76 Ø6 FF 76
[ØS5]  694: Ø4 BA 22 Ø6 FF D2 B8 ØC 7C 5Ø B8 67 1F 5Ø BA 86
[9W5]  695: Ø5 FF D2 8B EC FF 76 Ø6 FF 76 Ø4 BA 22 Ø6 FF D2
[PW5]  696: B8 DE 7C 5Ø B8 CB 53 5Ø BA 86 Ø5 FF D2 8B EC FF
[6N5]  697: 76 Ø6 FF 76 Ø4 BA 22 Ø6 FF D2 B8 14 7D 5Ø B8 7Ø
[7V5]  698: 64 5Ø BA 86 Ø5 FF D2 8B EC FF 76 Ø6 FF 76 Ø4 BA
[4S5]  699: 22 Ø6 FF D2 B8 B7 7D 5Ø B8 51 EA 5Ø BA 86 Ø5 FF
[SU5]  7ØØ: D2 8B EC FF 76 Ø6 FF 76 Ø4 BA 22 Ø6 FF D2 B8 63
[DT5]  7Ø1: 7D 5Ø B8 88 3Ø 5Ø BA 86 Ø5 FF D2 8B EC FF 76 Ø6
[AN5]  7Ø2: FF 76 Ø4 BA 22 Ø6 FF D2 B8 92 7E 5Ø B8 99 44 5Ø
[EV5]  7Ø3: BA 86 Ø5 FF D2 8B EC FF 76 Ø6 FF 76 Ø4 BA 22 Ø6
[ZX5]  7Ø4: FF D2 B8 4C 7E 5Ø B8 91 CC 5Ø BA 86 Ø5 FF D2 8B
[PT5]  7Ø5: EC FF 76 Ø6 FF 76 Ø4 BA 22 Ø6 FF D2 B8 AA 7F 5Ø
[ZY5]  7Ø6: B8 AA AA 5Ø BA 86 Ø5 FF D2 8B EC FF 76 Ø6 FF 76
[GF5]  7Ø7: Ø4 BA 22 Ø6 FF D2 B8 ØØ 81 5Ø 33 CØ 5Ø BA 86 Ø5
[CW5]  7Ø8: FF D2 8B EC 8F 46 Ø4 8F 46 Ø6 BA 22 Ø6 FF D2 8B
[TC5]  7Ø9: EC 8Ø 7E Ø9 81 72 19 8Ø 7E Ø3 ØØ 74 Ø4 8Ø 76 Ø2
[7R5]  71Ø: 8Ø B8 49 81 5Ø B8 DA ØF 5Ø BA 86 Ø5 FF D2 8B EC
[9C5]  711: F6 46 Ø8 8Ø 74 ØA 8Ø 7E Ø3 ØØ 74 Ø4 8Ø 76 Ø2 8Ø
[VL5]  712: 8F 46 Ø6 8F 46 Ø8 C3 Ø7 5B 8A 17 8B 42 33
[SG5]  713: C9 33 D2 B8 Ø1 42 CD 21 72 37 52 5Ø 33 C9 33 D2
[HS5]  714: B8 Ø2 42 CD 21 8B FØ 8B FA 5A 59 B8 ØØ 42 CD 21
[AS5]  715: 72 1F 8B CE 8B D7 B7 AØ 23 D2 78 Ø8 D1 E1 D1 D2
[6V5]  716: FE CF 75 F4 8A DE 8Ø E3 7F 53 8A E9 8A D5 51 Ø6
[8A2]  717: C3 33 CØ 5Ø 5Ø Ø6 C3
```

Listing 3. Die neue, assemblierte SBasic-Laufzeitbibliothek »sbasic.bin« als Hexdump

Der Bilderdieb

Nochmals verbesserte Snapshot-Version

Sie möchten sich Ihre eigene »Dia-Show« von tollen Grafiken aus Spielen und Anwendungsprogrammen zusammenstellen? Oder Sie wollen den aktuellen Bildschirminhalt zur späteren Verwendung komfortabel speichern? Mit dem residenten Hilfsprogramm »Snapshot« machen Sie auf Tastendruck aus jeder Grafik ein ausführbares Programm.

Zum ersten Mal wurde Snapshot in der Ausgabe 5 der DOS Extra veröffentlicht, kleinere Korrekturen folgten. Der Autor hat weitere, wichtige Verbesserungen angebracht, die wir Ihnen natürlich nicht vorenthalten wollen. Der Textkasten sagt Ihnen, was sich explizit geändert hat.

Snapshot (Listing 1) ist ein speicherresidentes Programm. Nach dem Aufruf wartet es darauf, vom Anwender mit der Tastenkombination [Alt-PrtSc] (beziehungsweise [Alt-Druck] auf deutschen Tastaturen) aufgerufen zu werden. Die eigentliche Aufgabe von Snapshot besteht darin, den momentan sichtbaren Bildschirm in komprimierter Form als ausführbare Exe-Datei auf Diskette oder Festplatte zu speichern und sich dann wieder »schlafen zu legen«.

Das hört sich einfach an, ist aber angesichts der vielen Grafikstandards (MDA, CGA, EGA, MCGA und VGA) ein wahres Verwirrspiel. Hinzu kommt, daß einige Computer (Schneider, Olivetti) und fast alle Kartennachbauer die vorgegebenen Möglichkeiten des Originals übertreffen wollten und zusätzliche Grafikmodi einführten, die natürlich zu nichts außer sich selbst kompatibel sind. Die Hercules-Karte ist ein bekanntes Beispiel hierfür.

Auch wenn sie sich inzwischen zum Pseudostandard für preiswerte Rechnerkonfigurationen entwickelt hat, so ist doch nicht zu übersehen, daß sie nicht von IBM selbst entwickelt wurde. Der Grafikmodus wird daher in keiner Weise vom IBM-kompatiblen BIOS unterstützt. Dadurch muß jedes Hercules-Grafikprogramm direkt auf die Register des Video-Controllers zugreifen, was nicht gerade der Kompatibilität solcher Programme zuträglich ist.

Diese Tatsachen haben dazu geführt, daß bei der Entwicklung des Snapshot-Programms viel Wert auf Erweiterbarkeit gelegt wurde. Sie können es somit auch mit Ihrer speziellen Grafikkarte verwenden. Die Sondermodi »Hercules« und »Schneider 2« sind allerdings bereits integriert, eben weil sie weit verbreitet sind. Und sollten Sie eine EVA/1024-VGA- oder dazu kompatible Karte besitzen, so müssen Sie sich um deren spezielle Modi ebenfalls nicht mehr kümmern.

Weil einige Programme (zum Beispiel Microsoft Windows) mit einem eigenen Tastaturtreiber arbeiten, wurde eine Routine in den Ticker-Interrupt eingehängt, die den Tastaturvektor ständig überwacht. Wenn ein Anwenderprogramm nun diesen Vektor verändert, so wird die Ta-

Was ist neu an der Version 2.7 von Snapshot?

Auch die letzte Korrektur von Snapshot in der Ausgabe 3'89 der DOS International hat noch nicht alle Fehler aufgedeckt. So gab es zum Beispiel Schwierigkeiten mit Bildern, die größer als 64 KByte waren. Dies lag daran, daß der Stack mit 64 Byte eindeutig zu klein geraten war. Als Folge davon kam es spätestens bei drei resident geladenen Hintergrund-Utilities zu einem Wrap-Around des Stackpointers, wodurch Teile der gepackten Bilddaten zerstört wurden. In der neuen Version beträgt der Stack deshalb 1024 Byte. Weil dies zum Speichern zuviel ist, wurde er hinter das eigentliche Bild gelegt. Allerdings ist nun wegen der veränderbaren Dateilänge eine etwas aufwendigere Berechnung notwendig.

Ein anderer Fehler hatte sich beim Mitprotokollieren der EGA-Farben im umgeleiteten Video-Interrupt 10hex eingeschlichen. Dort wurde die Nummer der Farbe dem CL-Register entnommen und nicht dem BL-Register, wie es richtig gewesen wäre. Und schließlich wurde der Startwert zum direkten Auslesen der VGA-Farben nicht in das Read-Adress-Register, sondern in das Write-Adress-Register des DAC geschrieben. Seltsamerweise jedoch funktionierte es trotzdem ...

Beseitigt wurden auch ein paar kleinere Ungereimtheiten, wie beispielsweise das Anzeigen eines Hercules-Bildes im Full-Modus (dadurch gibt es Schwierigkeiten mit zusätzlich eingebauten CGA-Karten) sowie das ungewöhnliche Abschalten des Cursors in den Textmodi, wofür eigentlich Bit 5 des Startwerts zuständig ist.

Die restlichen Änderungen betreffen keine Fehler, sondern fügen einen neuen Modus hinzu: die 16-Farben-Grafik der CGA-Karte bei einer Auflösung von 160x100 Bildpunkten. Programme, die diesen Modus verwenden, finden Sie in der DOS International 4'88 sowie in den Ausgaben 5'89 und 6'89 der DOS Extra.

Allerdings gibt es in diesen Programmen keinerlei Vorkehrungen, um den geänderten Modus residenten Hintergrundprogrammen kenntlich zu machen. Um aber jetzt nicht dem schlechten Beispiel der Hercules-Karte zu folgen (Erkennen des Modus nur durch Auslesen der Lightpen-Register, was bei VGA-Karten ohnehin nicht funktioniert), erkennt Snapshot diesen Spezialmodus nur, wenn er im BIOS-Datenbereich als 7Chex eingetragen wird. Fügen Sie also in Ihre entsprechenden Programme folgende Anweisung **nach** dem Einschalten der 16-Farben-Grafik ein:

```
DEF SEG=0 : POKE &H449,&H7C
```

Diese Zeile hat zudem den Vorteil, daß Sie damit auch anderen residenten Utilities die Änderung des Bildschirmmodus mitteilen und so weitere Komplikationen verhindern. Aber auch wenn Sie das entsprechende Programm nicht selbst geschrieben haben, können Sie Snapshot den Modus explizit mit

```
snapshot /7c
```

mitteilen. Von da an werden alle Bilder immer als 16-Farben-Grafik abgelegt, bis Sie mit

```
snapshot /a
```

die Moduserkennung wieder auf »Automatik« stellen.

Der 16-Farben-Modus funktioniert allerdings nicht immer, so gibt es zumeist Schwierigkeiten bei EGA- oder VGA-Karten. Wenn Sie einen solchen Adapter besitzen, dann müssen Sie vorher die Kompatibilität zum CGA-Modus auf Registerebene mit Hilfe eines (meist mitgelieferten) Umschaltprogramms sicherstellen (zum Beispiel bei Video Seven mit »v7vga cga:on« oder bei den Tseng-Karten mit »vmode cga«).

stenkombination [Alt-PrtSc] nicht mehr erkannt. In diesem Fall »drängelt« sich Snapshot einfach wieder an die erste Position der Reihe von Programmen, die den Vektor für sich beanspruchen.

Leider haben einige Programme, wie zum Beispiel der AFD-Debugger und Quick Basic, für solch ein Vorgehen kein Verständnis und stürzen umgehend ab. Allerdings kann auch der Interrupt 05hex, der das Betätigen von [Shift-PrtSc] aufruft, das Snapshot-Programm aktivieren. Geben Sie hierzu im Listing der Konstanten »intnr« den Wert 5 ein. Wenn Sie keinen MASM-Assembler ab Version 4.0 besitzen, dann können Sie das bereits übersetzte Programm (Listing 2) abtippen. In diesem Fall bleibt Ihnen jedoch die variable Interrupt-Anpassung verwehrt. Sie müssen sich mit den vorgegebenen Videomodi begnügen. Andernfalls generieren Sie aus der Quelldatei mit folgenden drei Befehlen die ausführbare Com-Datei:

```
masm snapshot;
link snapshot;
exe2bin snapshot snapshot.com;
```

Rufen Sie dann das Programm mit folgenden, optionalen Parametern auf:

```
snapshot <Dateiname> </Option> </Option> <...>
```

Der Dateiname kann Angaben über Laufwerk und Pfad enthalten und darf maximal 62 Zeichen umfassen. Unter diesem Namen wird der Bildschirm gespeichert, wenn Sie [Alt-PrtSc] drücken. Falls es die entsprechende Datei schon gab, wird sie ohne Rückfrage gelöscht. Fehlt diese Angabe beim Aufruf, dann speichert Snapshot die Bilder beginnend mit dem Dateinamen »bild0000.exe« (siehe Marke »pfad« in Listing 1) im aktuellen Verzeichnis.

Damit es nicht erforderlich ist, bei jedem Bild einen neuen Namen einzugeben, werden auch Ziffern, die sich im Namen oder in der Erweiterung befinden, bei jedem erfolgreichen Aufruf weitergezählt. Aus »c:\grafik00.exe« wird also »c:\grafik01.exe« und so weiter.

Leider halten sich nicht alle Programme an die »saubere« Modusumschaltung über das BIOS. Um in einem derartigen Fall nicht den Inhalt eines falschen Grafikbereichs zu speichern, erzwingen Sie einen bestimmen Modus per Schalter. Die Syntax dafür lautet:

```
/<Modus>[,<Seite>]
```

Der »Modus« steht für eine zweistellige Hexadezimalzahl von 00 bis 7Fhex. Die Bedeutung dieser Zahl entnehmen Sie bitte den Handbüchern zu Computer und Grafikkarte. Die Angabe der Seiten beginnt bei 0. Ein Beispiel verdeutlicht die Arbeit mit diesem Schalter: Angenommen, Sie besitzen eine EGA-Karte und wollen die Titelgrafik des Adventures »The Pawn« speichern. Der erste Versuch scheitert daran, daß Snapshot zwar ein sinnvolles Bild generiert, dieses aber keine Grafik, sondern nur den Schriftzug »Pondering...« enthält. Es liegt die Vermutung nahe, daß das Spiel »am BIOS vorbei« auf die zweite Grafikseite umgeschaltet, Snapshot aber den Inhalt der ersten Seite gespeichert hat. Aus der Auflösung des Bildes läßt mit etwas Erfahrung der verwendete Videomodus erkennen. Hier sind es 640x350 Punkte in 16 Farben, was dem EGA-Modus 10hex entspricht. Sie booten den Rechner neu und starten Snapshot also mit der Option »/10,1« (Videomodus 10hex, Seite 1) – und diesmal klappt es auch. Wollen Sie, daß die Erkennung des Videomodus wieder automatisch erfolgt, so rufen Sie Snapshot erneut, aber diesmal mit dem Schalter »/a« auf.

Ist Ihr Programm der HIT?

Der DMV-Verlag sucht ständig nach neuer, interessanter Software zur Aufnahme in unser Softwaresortiment.

Dabei ist es einerlei, ob Sie nun ein Anwendungs- oder ein Spielprogramm geschrieben haben.

Der DMV-Verlag bietet Ihnen sein Software-Know-how an!

Bitte Bestellkarte benutzen

DMV Verlag · Postfach 250 · 3440 Eschwege

Für Besitzer mancher Grafikkarten, die unter anderem auch eine Hercules-Emulation bieten (zum Beispiel eine an einen Monochrom-Monitor angeschlossene VGA-Karte), wurden ebenfalls zwei Schalter vorgesehen. Bei der Hercules-Karte läßt sich nämlich der aktuelle Modus (Text oder Grafik) nur durch Vergleichen des Lightpen-Registers mit einem bestimmten Schwellwert erkennen. Manche »pseudo-kompatiblen« Karten verzichten allerdings auf den Lightpen-Anschluß, so daß eine Unterscheidung nach der Lightpen-Methode gar nicht erst möglich ist.
Aber auch die Original-Hercules-Karte besitzt einen Nachteil: Es gibt keine Möglichkeit, die jeweils aktive Grafikseitennummer zu bestimmen. Sauber entwickelte Programme setzen zwar die entsprechenden Bit 1 und 7 der Adresse 0040:0065hex auf 1, aber darauf kann sich Snapshot natürlich nicht verlassen. Auch hier hilft nur ein weiterer Schalter. Dieser heißt

```
/h[,<Seite>]«
```

Ist er aktiviert, so speichert Snapshot bei eingeschaltetem Videomodus 07hex (Hercules) immer die von Ihnen gewünschte Grafikseite. Danach schalten Sie durch einen erneuten Aufruf von »snapshot.com«, diesmal mit dem Parameter »/m« (wie »MDA«), wieder auf automatische Hercules-Erkennung um.
Hin und wieder kann es vorkommen, daß nach dem Betätigen von [Alt-PrtSc] ein Ton ausgegeben wird. Dadurch möchte Ihnen Snapshot mitteilen, daß irgend etwas nicht funktioniert hat. Folgende Fehlerursachen sind möglich:

- Der momentan eingestelle Videomodus ist unbekannt,
- der angegebene Pfad ist fehlerhaft,
- es trat ein Schreibfehler auf (wird erst ab MS-DOS 3.0 abgefangen) oder
- auf der Diskette ist nicht mehr genügend Platz zum Speichern der Datei vorhanden.

In einigen der genannten Fälle wird zwar eine Exe-Datei angelegt, ihre Ausführung jedoch durch das Betriebssystem verhindert.

Für Assembler-Programmierer bietet »snapshot.asm« vielfältige Änderungsmöglichkeiten

Möchten Sie wissen, welche Videomodi Snapshot verarbeiten kann, so sehen Sie sich das Assembler-Listing an. Der erste Block gilt für die von IBM unterstützten Modi CGA, MDA, EGA, MCGA und VGA, wobei den Nummern 06 und 07 eine besondere Bedeutung zukommt. Sie stehen nämlich für jeweils zwei Modi: »CGA« und »Schneider 2« beziehungsweise »MDA« und »Hercules«. Im einzelnen müssen die korrespondierenden Unterroutinen selbst klären, welcher Modus denn nun wirklich aktiv ist. Von Interesse ist in diesem Zusammenhang die Erkennung des Schneider-Modus mit 640x200 Punkten in 16 Farben. So seltsam es auch klingen mag: Der Schneider-PC kennt überhaupt keinen CGA-Modus 06! Er befindet sich in diesem Fall immer im Grafikmodus 2, nur haben hierbei alle vier Bildebenen denselben Inhalt. Die Programmroutine muß also die Grafikebenen Byte für Byte vergleichen, um den Modus erkennen zu können.
Der zweite Block bei den Videomodi ist speziell für die EVA/1024-Karte gedacht. Besitzen Sie eine andere Karte, dann können Sie den Programmblock nebst zugehöriger Unterroutinen bedenkenlos löschen. Sie können auch die Unterprogramme für den Schneider- und den Hercules-Modus weglassen, müssen dann allerdings die Marken bei den Videomodi in »cga« ändern. Falls Sie die EGA- beziehungsweise VGA-Modi nicht benötigen, ist es zweckmäßig, zusätzlich zu den Unterprogrammen auch bei der Modusdefinition alle Zeilen mit einer entsprechenden Marke am Schluß zu löschen.
Wenn Sie neue Videomodi hinzufügen wollen, dann müssen Sie den Aufbau einer Definitionszeile kennen. Eingefügt wird diese vor der Marke »videoende« und hat folgenden Aufbau:

```
v <Modus>, <Segment>, <Seitenfaktor>, <Länge>, <Wortzugriff>, <Unterprogramm>
```

Der Wert »Modus« entspricht demjenigen beim Aufruf der BIOS-Funktionen 00 und 0Fhex. Das »Segment« steht für den Anfang der ersten Bildschirmseite. Den »Seitenfaktor« multipliziert Snapshot mit 256 und erkennt dadurch den Abstand (in Byte) von einer Bildschirmseite zur nächsten. Die »Länge« wird ebenfalls in Byte angegeben, wobei der Wert 0 als 65536 interpretiert wird. »Wortzugriff« ist ein Flag, das vor allem für die Effizienz des Packers verantwortlich ist. In den Textmodi hätte das byteweise Suchen nach sich wiederholenden Daten nämlich wenig Sinn, weil zwischen zwei Zeichen immer ein Attributbyte liegt. Der Packer sollte also nicht nach byteweisen, sondern wortweisen Wiederholungen suchen, was ihm mit diesem Flag mitgeteilt wird.
Der Parameter »Unterprogramm« schließlich ist für das eigentliche Speichern verantwortlich. An dieser Stelle können auch Unterscheidungen zwischen ansonsten gleichen Modi eingebaut werden (siehe »Schneider 2« und »Hercules«).
Die Namen der Unterprogramme lassen nicht unbedingt auf die verwendete Kartenemulation schließen, so wird zum Beispiel die Routine »cga« auch von dem VGA-Modus mit nur einer Grafikebene aufgerufen.
Im Prinzip brauchen Sie für die Entwicklung eigener Unterprogramme nur drei Routinen und deren Parameter zu kennen:
Die Routine »vorspann« wird nur einmal am Anfang aufgerufen. Sie verändert AX, DX und SI und benötigt in »neuermodus« den gewünschten Modus, in »videobasis« das Bildschirmsegment und in »wort« das Flag für den Wortzugriff. Diese drei Werte werden bereits vom Hauptprogramm aus der Definitionszeile gelesen und eingesetzt.
Die Routine »schreiben« schreibt einen Datenblock, der durch DS:SI adressiert ist, auf Diskette oder Festplatte. Das erste Wort dieses Blocks muß dessen Länge enthalten. AX und SI werden überschrieben.
Am wichtigsten ist die Routine »packen«. Sie schreibt den Block, der durch »videobasis:BX« adressiert ist und eine Länge von CX Byte besitzt, nebst einem Aufruf des Entpackers selbst auf Diskette. Das Hauptprogramm berechnet BX je nach aktueller Seite, CX übernimmt es direkt aus der Definitionszeile. Der Packer verändert AX und SI.
Die Register DS, ES und DI dürfen von Ihren Routinen nicht verändert werden, weil sie für den Variablenzugriff und für Schreibzugriffe auf den Datenträger benötigt werden. Die eigentliche Aufgabe Ihrer Routine besteht nun darin, zwischen diesen Unterprogrammaufrufen die einzelnen Ebenen der Grafikkarte zum Lesen vorzubereiten und die entsprechenden Umschaltbefehle für das spätere Entpacken auf Datenträger zu schreiben. Sind Sie damit fertig, dann kehren Sie mit »ret« zum Hauptprogramm zurück. Besitzt Ihre Grafikkarte allerdings nur eine Ebene, dann genügt bei den Videomodi eine Definition mit »cga« als Marke.
So umfangreich und komfortabel das BIOS der VGA-Karte auch sein mag, es stellt den Anwender manchmal vor schier unlösbare Probleme, wenn er bei angeschlossenem Schwarzweißmonitor die Farbpalette auslesen möchte. Denn wenn Sie zum Beispiel die Palette 0/0/0 setzen, dann erhalten Sie mit den entsprechenden Funktionen 1010hex und 1012hex die Folge 7/7/7 als Rückmeldung. Daran ändert auch das Abschalten des »Gray Scale Summing« nichts.

Aber auch bei Spielen kann es zu Problemen mit der Farbgestaltung kommen: Einige Spiele setzen im EGA-Betrieb die Farben nicht über die BIOS-Routinen, die von Snapshot abgefangen werden, so daß die tatsächlich eingestellten Werte unwiderruflich in den Tiefen des Video-Controllers verlorengehen. Für derartige Fälle ist das Programm »palette.pas« (Listing 3) zuständig. Es dient der Einstellung einzelner Farben bei den CGA-Vielfarbmodi 04 und 05 sowie allen EGA- und VGA-Grafikmodi außer 0Fhex (EGA-Monochrom). Übersetzen Sie »palette.pas« mit Turbo Pascal ab Version 4.0, und starten Sie es mit

```
palette <Bildname>
```

Der »Bildname« bestimmt eine zuvor mit Snapshot angelegte Datei. Die Erweiterung »exe« muß nicht angegeben werden.
Die Bedienung des Palette-Programms ist vom Videomodus abhängig. Bei CGA-Bildern können Sie mit der Taste [P] die Palette und mit [H] die Hintergrundfarbe auswählen. Anschließend speichern Sie das Bild entweder mit [Enter] auf Diskette oder Festplatte, oder Sie brechen das Programm mit [Escape] ab.
Bei EGA- und VGA-Bildern ist angesichts der Farbenvielfalt eine solch einfache Bedienung nicht mehr möglich. Hier müssen Sie zuerst mit den Cursor-Tasten [Links], [Rechts], [Home], [End] sowie den Tastenkombinationen [Ctrl-Links] und [Ctrl-Rechts] eine Farbe anwählen, die Sie im aktuellen Bild ändern möchten. Zur Orientierung blinkt die entsprechende Farbe kurz auf. Mit [Space] wiederholen Sie das Blinken. Die derart ausgewählte Farbe verändern Sie nun mit den Anfangsbuchstaben der englischen Grund- und Mischfarben, also:

```
[R] (»red«, Rot)
[G] (»green«, Grün)
[B] (»blue«, Blau)
[C] (»cyan«, Zyan)
[Y] (»yellow«, Gelb)
[M] (»magenta«, Magenta)
[W] (»white«, Weiß)
```

Diese Tasten addieren normalerweise den betreffenden Farbanteil zur Mischfarbe. Wenn Sie die jeweilige Taste jedoch zusammen mit [Shift] drücken, so wird die Farbe entfernt. Wollen Sie eine Farbe völlig neu einstellen, so setzen Sie sie vorher mit [0] auf Schwarz. Zum Schluß sichern Sie die mühsam erarbeitete Palette mit [Enter] auf Diskette.

Übrigens, mit Snapshot erzeugte Bildprogramme zeigen das Bild an und warten dann auf einen Tastendruck. Wollen Sie in eigenen Demonstrationsprogrammen mehrere Snapshots fortlaufend darstellen, aber den Benutzer nicht zu wiederholtem Tastendruck nötigen, dann rufen Sie die Bilddatei mit der Option »/c« auf. Das Programm wartet dann nicht auf eine Bestätigung, sondern kehrt nach der Generierung der Grafik sofort zur MS-DOS-Ebene zurück.

Der Betrachtungsgenuß eines Bildes, das nur eine halbe Sekunde auf dem Schirm zu sehen ist, hält sich wahrscheinlich in Grenzen. Ihr Programm oder die Stapeldatei hat deshalb selbst dafür zu sorgen, daß nachfolgende Darstellungen in einem anderen Videomodus wieder lesbar werden, zum Beispiel durch das Aufrufen eines Snapshot-Bildes in der dann gewünschten Auflösung.

(Wolfgang Lorenz/tr)

Sesam öffne dich!

BCi DisAsm erschließt jedes Programm!

Wandelt den Programmcode in ein dokumentiertes Assemblerlisting um. Mit normalen Editoren können Sie nun verändern und modifizieren, und mit gängigen Assemblern und Debuggern kann danach wieder assembliert werden.

Es gibt kaum einen, der BCi DisAsm das Wasser reichen kann!

BCi DisAsm i86
Intelligenter Disassembler Version 1.0

Auch Experten kennen nicht immer die Lösung, aber sie wissen, wo sie suchen müssen.
BCi DisAsm hilft JEDEM, die verwirrenden und frustrierenden Software-Puzzles, welche sich beim Umgang mit dem Computer ergeben, zu lösen. Und zwar schneller als mit jedem anderen Produkt, das Sie kaufen können, egal, um welche Preisklasse es sich handelt.
Mit dem BCi DisAsm erwerben Sie sofort ein Know-how, daß Sie normalerweise Jahre an Erfahrung kosten würde. Versuchen Sie nicht, das Rad neu zu erfinden!

BCi DisAsm – Der intelligente Disassembler
BCi DisAsm ist ein intelligenter, selbsttätig dokumentierender MS-DOS-Disassembler. Die ausgefeilten Routinen im BCi DisAsm analysieren in kürzester Zeit selbst komplexen und umfangreichen Programmcode und zeigen dabei potentielle Problempunkte auf.

BCi DisAsm – Das Lernmittel
Sie brauchen kein Spezialist in Assembler-Programmierung zu sein. BCi DisAsm produziert aus jedem beliebigen Programm ihres IBM-kompatiblen Computers ein gut dokumentiertes, leicht verständliches Assemblerlisting.

BCi DisAsm – Einfach in der Handhabung
BCi DisAsm kann sowohl über Parameter von der DOS-Kommandozeile aus als auch interaktiv über Menüs gesteuert werden. Die Menüstruktur ist Lotus-kompatibel und selbstverständlich in Deutsch. Es besteht keine Notwendigkeit, neue Befehle zu lernen oder ständig im Handbuch nachzuschlagen. BCi DisAsm ist unmittelbar anwendbar!

Minimale Systemanforderungen
256 kByte RAM, 8088/8086/80186/80286 oder 80386 CPU, MS-DOS oder PC-DOS Version 2.0 oder größer

BCi DisAsm wird Ihnen bei den folgenden Schritten helfen:

1. Lernen Sie Assembler-Programmierung, wenn Sie möchten.
2. Entdecken Sie, warum Programm XYZ auf Ihrem Rechner nicht läuft, und tun Sie etwas dagegen.
3. Ändern oder entfernen Sie Programmteile.
4. Sichern Sie sich gegen destruktive Programme ab. Finden Sie einen "Virus", und entfernen Sie ihn.
5. Integrieren Sie Code aus compilierten Programmen in eigene Anwendungen.
6. Erhöhen Sie die Kompatibilität Ihrer Software.
7. Passen Sie Software an verschiedene MS-DOS-Versionen an.

© 1987 BCi, West Germany und USA
Lotus ist eingetragenes Warenzeichen der Lotus Corp.
MS-DOS ist eingetragenes Warenzeichen der Microsoft Corp.

BCi DisAsm kostet inkl. eines sehr ausführlichen deutschen Handbuchs (mehr als 100 Seiten) nur

199,– DM (unverbindliche Preisempfehlung)
Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	199,– DM	Einzelpreis	199,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	203,– DM	Endpreis	205,– DM

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Name	snapshot.asm
Sprache	Assembler, MASM ab 4.Ø, TASM
Grafikkarte	jede
Besonderheiten	keine

```
[PC3]    1: code segment public
[8K3]    2: assume cs:code,ds:code
[3A1]    3: org Ø1ØØh
[ØØØ]    4:
[7V5]    5:     ;Interrupt 5 (PrtSc) oder 9 (KbdInt)
[4D1]    6: intnr = 9
[ØØØ]    7:
[2Ø1]    8: begin:
[SX2]    9:   jmp installieren
[ØØØ]   1Ø:
[2M7]   11:     ;Struktur für jeweils eine Definitionsz
           eile
[VA1]   12: v struc
[1K1]   13:   modus db ?
[6K1]   14:   basis dw ?
[7S1]   15:   faktor db ?
[8R1]   16:   laenge dw ?
[XE1]   17:   text db ?
[N12]   18:   routine dw ?
[FZØ]   19: v ends
[H2C]   2Ø:     ;Länge der Struktur in Bytes, die MASM-
           Funktion "Length" mag hier nicht
[SL2]   21: lengthvideo = 9
[ØØØ]   22:
[42B]   23:     ;Definition der von IBM unterstützten V
           ideomodi außer PCjr
[1T1]   24: videomodi:
[8M4]   25: v <ØØh,Øb8ØØh,8,2ØØØ,1,cgab> ;4Ø*25
[AM4]   26: v <Ø1h,Øb8ØØh,8,2ØØØ,1,cgab> ;4Ø*25
[6N4]   27: v <Ø2h,Øb8ØØh,16,4ØØØ,1,cgab> ;8Ø*25
[8N4]   28: v <Ø3h,Øb8ØØh,16,4ØØØ,1,cgab> ;8Ø*25
[985]   29: v <Ø4h,Øb8ØØh,64,16192,Ø,cgac> ;32Ø*2ØØ*4
[B85]   3Ø: v <Ø5h,Øb8ØØh,64,16192,Ø,cgac> ;32Ø*2ØØ*4
[MA6]   31: v <Ø6h,Øb8ØØh,64,16192,Ø,schneider> ;64Ø*2Ø
           Ø*2
[PE5]   32: v <Ø7h,ØbØØØh,16,4ØØØ,1,hercules> ;8Ø*25
[Y35]   33: v <Ødh,ØaØØØh,32,8ØØØ,Ø,ega> ;32Ø*2ØØ*16
[W65]   34: v <Øeh,ØaØØØh,64,16ØØØ,Ø,ega> ;64Ø*2ØØ*16
[B65]   35: v <Øfh,ØaØØØh,128,28ØØØ,Ø,ega> ;64Ø*35Ø*4
[W95]   36: v <1Øh,ØaØØØh,128,28ØØØ,Ø,ega> ;64Ø*35Ø*16
[D25]   37: v <11h,ØaØØØh,Ø,384ØØ,Ø,cga> ;64Ø*48Ø*2
[545]   38: v <12h,ØaØØØh,Ø,384ØØ,Ø,ega> ;64Ø*48Ø*16
[475]   39: v <13h,ØaØØØh,Ø,64ØØØ,Ø,vga> ;32Ø*2ØØ*256
[CC5]   4Ø: v <7ch,Øb8ØØh,Ø,32ØØØ,1,cga16> ;16Ø*1ØØ*16
[ØØØ]   41:
[Z56]   42:     ;Definition der Videomodi der EVA/1Ø24
[UR4]   43: v <18h,ØbØØØh,64,11616,1,cga> ;132*44
[3N4]   44: v <19h,ØbØØØh,16,66ØØ,1,cga> ;132*25
[TN4]   45: v <1ah,ØbØØØh,16,7392,1,cga> ;132*27
[7S4]   46: v <22h,Øb8ØØh,64,11616,1,cga> ;132*44
[MN4]   47: v <23h,Øb8ØØh,32,66ØØ,1,cga> ;132*25
[WN4]   48: v <24h,Øb8ØØh,32,7392,1,cga> ;132*27
[C45]   49: v <25h,ØaØØØh,Ø,384ØØ,Ø,ega> ;64Ø*48Ø*16
[1P4]   5Ø: v <26h,Øb8ØØh,64,96ØØ,1,cga> ;8Ø*6Ø
[C25]   51: v <29h,ØaØØØh,Ø,6ØØØØ,Ø,ega> ;8ØØ*6ØØ*4
[UN4]   52: v <2ah,Øb8ØØh,32,8ØØØ,1,cga> ;1ØØ*4Ø
[HT4]   53: v <2dh,ØaØØØh,Ø,Ø,Ø,eva> ;64Ø*35Ø*256
[UT4]   54: v <2eh,ØaØØØh,Ø,Ø,Ø,eva> ;64Ø*48Ø*256
[5R4]   55: v <37h,ØaØØØh,Ø,Ø,Ø,ega> ;1Ø24*768*16
[ØØØ]   56:
[4F6]   57:     ;Hier können Sie eigene Modi einfügen
[ØØØ]   58:
[BØ2]   59: videoende = $
[ØØØ]   6Ø:
[SV5]   61:     ;Blinken ein/aus für EGA-Textmodi
[NSØ]   62: cgab:
[T82]   63:   call vorspann
[KB3]   64:   mov si,offset blink
[ZC2]   65:   call schreiben
[9S1]   66:   jmp packen
[ØØØ]   67:
[7N3]   68: blink dw blinke-blinka
[V81]   69: blinka:
[EV1]   7Ø:   mov ax,1ØØ3h
[8A1]   71:   mov bl,1
[8A3]   72: blinken = byte ptr $-1
[Y71]   73:   int 1Øh
[8D1]   74: blinke = $
[ØØØ]   75:
[GR7]   76:     ;Einfache Modi ohne Farbpaletten und Bi
           tplanes
[TMØ]   77: cga:
[T82]   78:   call vorspann
[9S1]   79:   jmp packen
[ØØØ]   8Ø:
[S24]   81:     ;CGA-Modi 32Ø*2ØØ in 4 Farben
[PSØ]   82: cgac:
[T82]   83:   call vorspann
[6P3]   84:   mov si,offset cgafarbe
[ZC2]   85:   call schreiben
[9S1]   86:   jmp packen
[ØØØ]   87:
[FK3]   88: cgafarbe dw cgafe-cgaf
[TSØ]   89: cgaf:
[3V2]   9Ø:     ;Randfarbe setzen
[NM1]   91:   mov ah,Øbh
[SE1]   92:   mov bx,16
[183]   93: cgarand = byte ptr $-2
[Y71]   94:   int 1Øh
[DT2]   95:     ;Palette setzen
[NM1]   96:   mov ah,Øbh
[TL1]   97:   mov bx,256
[KS3]   98: cgapalette = byte ptr $-2
[Y71]   99:   int 1Øh
[B71]  1ØØ: cgafe = $
[ØØØ]  1Ø1:
[1B7]  1Ø2:     ;Prüfen ob Schneidermodus 2 oder CGA-Mo
           dus Ø6
[9R1]  1Ø3: schneider:
[TC1]  1Ø4:   push bx
[LH1]  1Ø5:   xor bx,bx
[XL1]  1Ø6:   mov ah,Ø6h
[D81]  1Ø7:   int 15h
[GWC]  1Ø8:     ;Bei Schneider-Computern enthält BX nac
           h diesem Interrupt die BIOS-Version
[AE1]  1Ø9:   and bx,bx
[T31]  11Ø:   pop bx
[RVØ]  111:   je cga
[37A]  112:     ;Die vier Bitplanes des Schneider PC au
           f Gleichheit prüfen
[TC1]  113:   push bx
[WC1]  114:   push cx
[BC1]  115:   push ds
[9Y2]  116:   mov ds,videobasis
[2X3]  117:     ;Port für Lesezugriffe
[XZ1]  118:   mov dx,Ø3deh
[VG1]  119:   xor si,si
[SU1]  12Ø:   mov cx,8Ø96
[X52]  121: pruefewort:
[1Y3]  122:     ;Wert von Seite Ø lesen
[5A1]  123:   mov al,Ø
[6H1]  124:   out dx,al
[VH2]  125:   mov bx,ds:[si]
[AY3]  126:     ;Mit Seite 1 vergleichen
[EØ1]  127:   inc al
[6H1]  128:   out dx,al
[YH2]  129:   cmp ds:[si],bx
[YW4]  13Ø:     ;Ungleich => Schneidermodus 2
[R73]  131:   jne schneidermodus
[BY3]  132:     ;Mit Seite 2 vergleichen
[EØ1]  133:   inc al
[6H1]  134:   out dx,al
[YH2]  135:   cmp ds:[si],bx
[R73]  136:   jne schneidermodus
[P71]  137:     ;Seite 3
[EØ1]  138:   inc al
[6H1]  139:   out dx,al
[PWØ]  14Ø:   lodsw
[TF1]  141:   cmp ax,bx
[UW2]  142:   loope pruefewort
[DT2]  143: schneidermodus:
[E31]  144:   pop ds
[V31]  145:   pop cx
[T31]  146:   pop bx
[RVØ]  147:   je cga
[S56]  148:     ;Bild im Schneidermodus 2 abspeichern
[T82]  149:   call vorspann
[B54]  15Ø:   mov si,offset schneiderØ
[ZC2]  151:   call schreiben
[K62]  152:     ;Lesen Seite Ø
[XZ1]  153:   mov dx,Ø3deh
[5A1]  154:   mov al,Ø
[6H1]  155:   out dx,al
[TY1]  156:   call packen
[C54]  157:   mov si,offset schneider1
[ZC2]  158:   call schreiben
[N62]  159:     ;Lesen Seite 1
[6A1]  16Ø:   mov al,1
[6H1]  161:   out dx,al
[TY1]  162:   call packen
[D54]  163:   mov si,offset schneider2
[ZC2]  164:   call schreiben
[S62]  165:     ;Lesen Seite 2
[7A1]  166:   mov al,2
[6H1]  167:   out dx,al
[TY1]  168:   call packen
[E54]  169:   mov si,offset schneider3
[ZC2]  17Ø:   call schreiben
[V62]  171:     ;Lesen Seite 3
[8A1]  172:   mov al,3
[6H1]  173:   out dx,al
[9S1]  174:   jmp packen
[ØØØ]  175:
[6F4]  176: schneiderØ dw schneider1-sØ
[YGØ]  177: sØ:
[ØR5]  178:     ;Alle 4 Bitplanes werden angezeigt
[DY1]  179:   mov dx,Ø3d9h
```

```
[9N1]  18Ø:    mov al,Øfh
[6H1]  181:    out dx,al
[DX2]  182:      ;Schreiben Seite Ø
[K71]  183:    add dx,4
[6A1]  184:    mov al,1
[6H1]  185:    out dx,al
[EF4]  186: schneider1 dw schneider2-s1
[1HØ]  187: s1:
[GX2]  188:      ;Schreiben Seite 1
[7A1]  189:    mov al,2
[6H1]  19Ø:    out dx,al
[PF4]  191: schneider2 dw schneider3-s2
[4HØ]  192: s2:
[LX2]  193:      ;Schreiben Seite 2
[9A1]  194:    mov al,4
[6H1]  195:    out dx,al
[FG4]  196: schneider3 dw schneidere-s3
[7HØ]  197: s3:
[PX2]  198:      ;Schreiben Seite 3
[DA1]  199:    mov al,8
[6H1]  2ØØ:    out dx,al
[L32]  2Ø1: schneidere = $
[ØØØ]  2Ø2:
[T47]  2Ø3:      ;Unterscheidung zwischen Hercules und M
       DA
[LK1]  2Ø4: hercules:
[A57]  2Ø5:      ;Vom Benutzer erzwungener Herculesmodus
[H54]  2Ø6:    cmp byte ptr hercmodus,Ø
[WC1]  2Ø7:    jne herc
[654]  2Ø8:    mov byte ptr hercseite,Ø
[G97]  2Ø9:      ;Bit 1 und 7 der BIOS-Variable 4Ø:65h a
       uswerten
[BC1]  21Ø:    push ds
[5N1]  211:    mov ax,4Øh
[1H1]  212:    mov ds,ax
[5S2]  213:    mov al,ds:[65h]
[E31]  214:    pop ds
[1S3]  215:      ;Bit 1: Ø=Text, 1=Grafik
[UE1]  216:    test al,2
[YZ1]  217:    je hercules?
[S44]  218:      ;Bit 7: Ø=Seite Ø, 1=Seite 1
[EU1]  219:    test al,128
[R51]  22Ø:    je herc
[754]  221:    mov byte ptr hercseite,1
[1E2]  222:    jmp short herc
[DS6]  223:      ;Herculeskarte auf Hardwarebasis suchen
[BS1]  224: hercules?:
[LB4]  225:      ;Lightpen-Flipflop setzen
[7Y1]  226:    mov dx,Ø3b9h
[6H1]  227:    out dx,al
[E35]  228:      ;Bit 2 des Status-Port gesetzt?
[S11]  229:    inc dx
[161]  23Ø:    in al,dx
[UE1]  231:    test al,2
[WY5]  232:      ;Nein, dann kein Lightpen vorhanden
[VVØ]  233:    je mda
[GM4]  234:      ;Lightpen-Flipflop löschen
[S11]  235:    inc dx
[6H1]  236:    out dx,al
[3V2]  237:      ;Bit 2 gelöscht?
[LØ1]  238:    dec dx
[161]  239:    in al,dx
[UE1]  24Ø:    test al,2
[H36]  241:      ;Ja, dann Text-/Grafikmodus prüfen
[7D1]  242:    je warteØ
[4NØ]  243: mda:
[U91]  244:    jmp cga
[NM5]  245:      ;Warten auf Rasterstrahl im Bild
[F81]  246: warteØ:
[161]  247:    in al,dx
[EU1]  248:    test al,128
[7D1]  249:    je warteØ
[2U5]  25Ø:      ;Warten auf Rücklauf des Strahls
[K81]  251: wartel:
[161]  252:    in al,dx
[EU1]  253:    test al,128
[WP1]  254:    jne wartel
[FD3]  255:      ;Flipflop set & reset
[S11]  256:    inc dx
[6H1]  257:    out dx,al
[Z91]  258:    sub dx,2
[6H1]  259:    out dx,al
[AX3]  26Ø:      ;Lightpenadresse lesen
[LX1]  261:    mov dx,Ø3b4h
[AR9]  262:    mov al,16 ;<-- hier eventuell 17 einsetze
       n (Peacock Hercules)
[6H1]  263:    out dx,al
[S11]  264:    inc dx
[161]  265:    in al,dx
[2F1]  266:    mov ah,al
[LØ1]  267:    dec dx
[9N6]  268:    mov al,17 ;<-- hier eventuell 16 einsetze
       n
[6H1]  269:    out dx,al
[S11]  27Ø:    inc dx
[161]  271:    in al,dx
[BX4]  272:      ;Mit Schwellwert vergleichen
[AT1]  273:    cmp ax,2958
[KVØ]  274:    jb mda
[YUØ]  275: herc:
[C77]  276:      ;Grafik => Paker auf Bytezugriff umstel
       len
[MD3]  277:    mov byte ptr wort,Ø
[T82]  278:    call vorspann
[B14]  279:    mov si,offset herculesØ
[ZC2]  28Ø:    call schreiben
[YT2]  281:      ;BX := Seite * 32768
[PT2]  282:    mov al,hercseite
[6K1]  283:    mov ah,128
[321]  284:    mul ah
[4F1]  285:    mov bh,al
[8W1]  286:    mov cx,32494
[9S1]  287:    jmp packen
[ØØØ]  288:
[C84]  289:      ;Grafikmodus einschalten
[ZV2]  29Ø: herculesØ dw he-hØ
[8GØ]  291: hØ:
[AR4]  292:      ;Karte in Half-Modus schalten
[VZ1]  293:    mov dx,Ø3bfh
[6A1]  294:    mov al,1
[6H1]  295:    out dx,al
[151]  296:    call h1
[G69]  297:      ;Tabelle der Werte für den 6845 (Hi=Wer
       t, Lo=RegisterNr)
[VØ5]  298:    dw 35ØØh,2dØ1h,2eØ2h,Ø7Ø3h,5bØ4h,Ø2Ø5h
[315]  299:    dw 57Ø6h,57Ø7h,Ø2Ø8h,Ø3Ø9h,ØØØah,ØØØbh
[BGØ]  3ØØ: h1:
[E31]  3Ø1:    pop si
[5A3]  3Ø2:      ;6845 initialisieren
[KE1]  3Ø3:    mov cx,12
[LX1]  3Ø4:    mov dx,Ø3b4h
[EGØ]  3Ø5: h2:
[PWØ]  3Ø6:    lodsw
[YH1]  3Ø7:    out dx,ax
[H71]  3Ø8:    loop h2
[8F5]  3Ø9:      ;Controller auf Grafik schalten
[3Y1]  31Ø:    mov dx,Ø3b8h
[6D1]  311:    mov al,1Ø
[6H1]  312:    out dx,al
[8PØ]  313: he = $
[ØØØ]  314:
[PF6]  315:      ;Unterprogramm für EGA/VGA mit 16 Farbe
       n
[XMØ]  316: ega:
[T82]  317:    call vorspann
[KX7]  318:      ;Im monochromen Modus ØFh gibt es keine
        Palette
[363]  319:    cmp neuermodus,Øfh
[3A2]  32Ø:    je zweiteebene
[6Ø4]  321:    mov si,offset egafarben
[ZC2]  322:    call schreiben
[552]  323: zweiteebene:
[3Z2]  324:    mov si,offset egaØ
[ZC2]  325:    call schreiben
[4U5]  326:      ;Indexregister auf Read-Map-Select
[UZ1]  327:    mov dx,Ø3ceh
[9A1]  328:    mov al,4
[6H1]  329:    out dx,al
[K62]  33Ø:      ;Lesen Seite Ø
[S11]  331:    inc dx
[5A1]  332:    mov al,Ø
[6H1]  333:    out dx,al
[TY1]  334:    call packen
[849]  335:      ;Der monochrome Modus verwendet nur die
        Seiten Ø und 2
[363]  336:    cmp neuermodus,Øfh
[BP2]  337:    je uebersprinqe1
[6Z2]  338:    mov si,offset ega1
[ZC2]  339:    call schreiben
[N62]  34Ø:      ;Lesen Seite 1
[6A1]  341:    mov al,1
[6H1]  342:    out dx,al
[TY1]  343:    call packen
[2F2]  344: uebersprinqe1:
[VX9]  345:      ;Der EVA-Spezialmodus 8ØØ*6ØØ*4 verwend
       et nur die Seiten Ø und 1
[B53]  346:    cmp neuermodus,29h
[DP2]  347:    je uebersprinqe3
[9Z2]  348:    mov si,offset ega2
[ZC2]  349:    call schreiben
[S62]  35Ø:      ;Lesen Seite 2
[7A1]  351:    mov al,2
[6H1]  352:    out dx,al
[TY1]  353:    call packen
[363]  354:    cmp neuermodus,Øfh
[DP2]  355:    je uebersprinqe3
[CZ2]  356:    mov si,offset ega3
[ZC2]  357:    call schreiben
[V62]  358:      ;Lesen Seite 3
[8A1]  359:    mov al,3
[6H1]  36Ø:    out dx,al
[TY1]  361:    call packen
[6F2]  362: uebersprinqe3:
[DE1]  363:    and cx,cx
[PY1]  364:    jne egaende
```

Listing 1. »snapshot.asm« ist ein ausgereiftes und nützliches Hilfsprogramm für jeden Grafikfan

```
[853]  365:    cmp neuermodus,37h
[PY1]  366:    jne egaende
[9VB]  367:      ;Beim EVA-Spezialmodus 1Ø24*768*16 ist
       eine Seite 96 KByte groß und kann
[4B7]  368:      ;nicht in einem Rutsch abgespeichert we
       rden
[NC3]  369:    mov videobasis,Øb000h
[9W1]  37Ø:    mov cx,32768
[LT3]  371:    mov si,offset evahires
[ZC2]  372:    call schreiben
[6U2]  373:    jmp zweiteebene
[YA1]  374: egaende:
[3PØ]  375:    ret
[ØØØ]  376:
[DT2]  377:      ;Palette setzen
[LM3]  378: egafarben dw egaØ-egafØ
[RXØ]  379: egafØ:
[GK1]  38Ø:    call egaf1
[CF3]  381: egapalette db 17 dup (Ø)
[TXØ]  382: egaf1:
[X31]  383:    pop dx
[EC1]  384:    push es
[8C1]  385:    push cs
[G31]  386:    pop es
[AV1]  387:    mov ax,1ØØ2h
[Y71]  388:    int 1Øh
[G31]  389:    pop es
[SC2]  39Ø: egaØ dw ega1-eØ
[2GØ]  391: eØ:
[5V4]  392:      ;Indexregister auf Map-Select
[PX1]  393:    mov dx,Ø3c4h
[7A1]  394:    mov al,2
[6H1]  395:    out dx,al
[DX2]  396:      ;Schreiben Seite Ø
[S11]  397:    inc dx
[6A1]  398:    mov al,1
[6H1]  399:    out dx,al
[YC2]  4ØØ: ega1 dw ega2-e1
[5GØ]  4Ø1: e1:
[GX2]  4Ø2:      ;Schreiben Seite 1
[7A1]  4Ø3:    mov al,2
[6H1]  4Ø4:    out dx,al
[4D2]  4Ø5: ega2 dw ega3-e2
[8GØ]  4Ø6: e2:
[LX2]  4Ø7:      ;Schreiben Seite 2
[9A1]  4Ø8:    mov al,4
[6H1]  4Ø9:    out dx,al
[U63]  41Ø: ega3 dw evahires-e3
[BGØ]  411: e3:
[PX2]  412:      ;Schreiben Seite 3
[DA1]  413:    mov al,8
[6H1]  414:    out dx,al
[ØA3]  415: evahires dw egae-eh
[AKØ]  416: eh:
[CR7]  417:      ;Für EVA 1Ø24*768*16 Bildschirmsegment
       neu laden
[XZ1]  418:    mov ax,Øb000h
[3H1]  419:    mov es,ax
[321]  42Ø: egae = $
[ØØØ]  421:
[ØD5]  422:      ;Unterprogramm für VGA-Modus 13h
[ZNØ]  423: vga:
[T82]  424:    call vorspann
[MR2]  425:    call vgapalette
[8S3]  426:      ;Auf SCQ-Modus prüfen
[BC1]  427:    push ds
[GH1]  428:    xor ax,ax
[1H1]  429:    mov ds,ax
[BC4]  43Ø:    cmp byte ptr ds:[484h],18h
[E31]  431:    pop ds
[5B1]  432:    jne scq
[9S1]  433:    jmp packen
[GØ5]  434:      ;Bild im SCQ-Modus abspeichern
[DRØ]  435: scq:
[XU1]  436:    mov cx,432ØØ
[UZ1]  437:    mov dx,Ø3ceh
[5M1]  438:    mov al,Ø4h
[6H1]  439:    out dx,al
[S11]  44Ø:    inc dx
[XT2]  441:      ;Schreibe Seite 2
[L93]  442:    mov si,offset scqdØ
[ZC2]  443:    call schreiben
[7A1]  444:    mov al,2
[6H1]  445:    out dx,al
[TY1]  446:    call packen
[VT2]  447:      ;Schreibe Seite 1
[R93]  448:    mov si,offset scqd1
[ZC2]  449:    call schreiben
[6A1]  45Ø:    mov al,1
[6H1]  451:    out dx,al
[TY1]  452:    call packen
[ZT2]  453:      ;Schreibe Seite 3
[V93]  454:    mov si,offset scqd2
[ZC2]  455:    call schreiben
[8A1]  456:    mov al,3
[6H1]  457:    out dx,al
[TY1]  458:    call packen
[TT2]  459:      ;Schreibe Seite Ø
```

```
[Z93]  46Ø:    mov si,offset scqd3
[ZC2]  461:    call schreiben
[5A1]  462:    mov al,Ø
[6H1]  463:    out dx,al
[9S1]  464:    jmp packen
[ØØØ]  465:
[8Z2]  466: scqdØ dw scqd1-scdØ
[3TØ]  467: scdØ:
[VD1]  468:    call scqØ
[FA3]  469:    db 45,Ø,59,8,Ø,48,117,1,1,15
[7F4]  47Ø:    db Ø,6,3,Ø,1Ø9,89,9Ø,142,98,139,11,62
[Ø94]  471:    db Ø,64,Ø,Ø,Ø,Ø,Ø,Ø,234,14Ø,223,45,Ø
[EG4]  472:    db 231,4,195,255,Ø,Ø,Ø,Ø,Ø,64,5,15,255
[YH1]  473: wortaus:
[TM1]  474:    xchg al,ah
[6H1]  475:    out dx,al
[S11]  476:    inc dx
[TM1]  477:    xchg al,ah
[6H1]  478:    out dx,al
[LØ1]  479:    dec dx
[2Ø1]  48Ø:    inc ah
[3PØ]  481:    ret
[RUØ]  482: scqØ:
[E31]  483:    pop si
[EC1]  484:    push es
[3B1]  485:    push di
[GH1]  486:    xor ax,ax
[3H1]  487:    mov es,ax
[KW1]  488:    mov di,Ø44ah
[ØZØ]  489:    movsw
[AV1]  49Ø:    mov di,Ø484h
[PXØ]  491:    movsb
[ØZØ]  492:    movsw
[UW1]  493:    mov di,Ø44ch
[ØZØ]  494:    movsw
[G21]  495:    pop di
[G31]  496:    pop es
[EMØ]  497:    cli
[PX1]  498:    mov dx,Ø3c4h
[T91]  499:    mov ah,Ø
[SUØ]  5ØØ: scq1:
[CVØ]  5Ø1:    lodsb
[A52]  5Ø2:    call wortaus
[C81]  5Ø3:    cmp ah,5
[A31]  5Ø4:    jb scq1
[7W1]  5Ø5:    mov dl,Øc2h
[3L1]  5Ø6:    mov al,199
[6H1]  5Ø7:    out dx,al
[DW1]  5Ø8:    mov dl,Øc4h
[PWØ]  5Ø9:    lodsw
[A52]  51Ø:    call wortaus
[6PØ]  511:    sti
[FW1]  512:    mov dl,Ød4h
[SE1]  513:    mov ax,17
[6H1]  514:    out dx,al
[S11]  515:    inc dx
[161]  516:    in al,dx
[DK1]  517:    and al,7fh
[6H1]  518:    out dx,al
[LØ1]  519:    dec dx
[TUØ]  52Ø: scq2:
[CVØ]  521:    lodsb
[A52]  522:    call wortaus
[LB1]  523:    cmp ah,25
[E31]  524:    Jb scq2
[UX1]  525:    mov dl,Øcch
[5A1]  526:    mov al,Ø
[6H1]  527:    out dx,al
[MX1]  528:    mov dl,Øcah
[AB1]  529:    mov ax,1
[6H1]  53Ø:    out dx,al
[ØY1]  531:    mov dl,Øceh
[UUØ]  532: scq3:
[CVØ]  533:    lodsb
[A52]  534:    call wortaus
[G81]  535:    cmp ah,9
[K31]  536:    jb scq3
[HZ2]  537: scqd1 dw scqd2-scd1
[4TØ]  538: scd1:
[PX1]  539:    mov dx,Ø3c4h
[7A1]  54Ø:    mov al,2
[6H1]  541:    out dx,al
[S11]  542:    inc dx
[8A1]  543:    mov al,3
[6H1]  544:    out dx,al
[UZ2]  545: scqd2 dw scqd3-scd2
[5TØ]  546: scd2:
[DA1]  547:    mov al,8
[6H1]  548:    out dx,al
[3Ø3]  549: scqd3 dw scqd4-scd3
[6TØ]  55Ø: scd3:
[6A1]  551:    mov al,1
[6H1]  552:    out dx,al
[V81]  553: scqd4 = $
[ØØØ]  554:
[3R4]  555:      ;CGA-Grafikmodus in 16 Farben
[AXØ]  556: cga16:
[BE6]  557:      ;Videomodus 7Ch verkraftet BIOS nicht
```

```
[MW2]  558:    mov neuermodus,3
[T82]  559:    call vorspann
[AA3]  560:    mov si,offset mode16
[ZC2]  561:    call schreiben
[9S1]  562:    jmp packen
[N23]  563: mode16 dw end16-anf16
[FY0]  564: anf16:
[SY1]  565:    call getadr
[EG4]  566:      ;Parametersatz für den 6845
[694]  567:    db 113,80,90,10,127,6,100,112,0,1,32
[MA1]  568: getadr:
[E31]  569:    pop si
[TX1]  570:    mov dx,03d4h
[0F1]  571:    xor ah,ah
[N81]  572: set6845:
[6F1]  573:    mov al,ah
[6H1]  574:    out dx,al
[S11]  575:    inc dx
[CV0]  576:    lodsb
[6H1]  577:    out dx,al
[L01]  578:    dec dx
[201]  579:    inc ah
[AB1]  580:    cmp ah,11
[AE1]  581:    jb set6845
[Y75]  582:      ;Reverse-Hilight statt Blinken
[VW1]  583:    mov dl,0d8h
[EA1]  584:    mov al,9
[6H1]  585:    out dx,al
[G37]  586:      ;Videomodus residenten PRGs bekanntgebe
       n
[EC1]  587:    push es
[GH1]  588:    xor ax,ax
[3H1]  589:    mov es,ax
[NS4]  590:    mov byte ptr es:[0449h],7ch
[G31]  591:    pop es
[X61]  592: end16 = $
[000]  593:
[DP6]  594:      ;VGA-Palette lesen und auf Disk schreib
       en
[202]  595: vgapalette:
[713]  596:    mov si,offset vga0
[ZC2]  597:    call schreiben
[WC1]  598:    push cx
[2Y1]  599:    mov dx,03c7h
[MF1]  600:    xor al,al
[6H1]  601:    out dx,al
[S11]  602:    inc dx
[S11]  603:    inc dx
[VL1]  604:    mov cx,256
[412]  605: schreibedac:
[161]  606:    in al,dx
[8N2]  607:    call schreibeal
[161]  608:    in al,dx
[8N2]  609:    call schreibeal
[161]  610:    in al,dx
[8N2]  611:    call schreibeal
[MS2]  612:    loop schreibedac
[V31]  613:    pop cx
[A13]  614:    mov si,offset vga1
[A82]  615:    jmp schreiben
[000]  616:
[6F3]  617:      ;VGA-Palette setzen
[FG2]  618: vga0 dw vga1-v0
[VB1]  619:      ;Jmp $+768
[2S1]  620: v0 db 0e8h,0,3
[8H2]  621: vga1 dw vgae-v1
[7H0]  622: v1:
[X31]  623:    pop dx
[EC1]  624:    push es
[8C1]  625:    push cs
[G31]  626:    pop es
[DV1]  627:    mov ax,1012h
[PKA]  628:      ;BIOS-Funktion: Setze Block von Palette
       nregistern mit [ES:DX]
[LH1]  629:    xor bx,bx
[VL1]  630:    mov cx,256
[Y71]  631:    int 10h
[G31]  632:    pop es
[531]  633: vgae = $
[000]  634:
[PD8]  635:      ;Unterprogramm für die beiden EVA-Modi
       mit 256 Farben
[AP0]  636: eva:
[T82]  637:    call vorspann
[MR2]  638:    call vgapalette
[KZ2]  639:    mov si,offset eva0
[ZC2]  640:    call schreiben
[K62]  641:      ;Lesen Seite 0
[PZ1]  642:    mov dx,03cdh
[G7A]  643:      ;Bit 0..2 Schreibseite, Bit 3..5 Lesese
       ite, Bit 6 immer gesetzt
[1M1]  644:    mov al,40h
[6H1]  645:    out dx,al
[TY1]  646:    call packen
[NZ2]  647:    mov si,offset eva1
[ZC2]  648:    call schreiben
[N62]  649:      ;Lesen Seite 1
[HM1]  650:    mov al,48h
[6H1]  651:    out dx,al
[TY1]  652:    call packen
[SZ2]  653:    mov si,offset eva2
[ZC2]  654:    call schreiben
[S62]  655:      ;Lesen Seite 2
[2M1]  656:    mov al,50h
[6H1]  657:    out dx,al
[TY1]  658:    call packen
[VZ2]  659:    mov si,offset eva3
[ZC2]  660:    call schreiben
[V62]  661:      ;Lesen Seite 3
[KM1]  662:    mov al,58h
[6H1]  663:    out dx,al
[163]  664:    cmp neuermodus,2dh
[TN1]  665:    jne kein2d
[CW1]  666:    mov cx,27392
[9S1]  667:    jmp packen
[F71]  668: kein2d:
[TY1]  669:    call packen
[YZ2]  670:    mov si,offset eva4
[ZC2]  671:    call schreiben
[Y62]  672:      ;Lesen Seite 4
[3M1]  673:    mov al,60h
[6H1]  674:    out dx,al
[363]  675:    cmp neuermodus,2eh
[WN1]  676:    jne kein2e
[NV1]  677:    mov cx,45056
[9S1]  678:    jmp packen
[K71]  679: kein2e:
[3P0]  680:    ret
[000]  681:
[UN2]  682: eva0 dw eva1-ev0
[6M0]  683: ev0:
[DX2]  684:      ;Schreiben Seite 0
[PZ1]  685:    mov dx,03cdh
[1M1]  686:    mov al,40h
[6H1]  687:    out dx,al
[XN2]  688: eva1 dw eva2-ev1
[AM0]  689: ev1:
[GX2]  690:      ;Schreiben Seite 1
[3M1]  691:    mov al,41h
[6H1]  692:    out dx,al
[0P2]  693: eva2 dw eva3-ev2
[EM0]  694: ev2:
[LX2]  695:      ;Schreiben Seite 2
[5M1]  696:    mov al,42h
[6H1]  697:    out dx,al
[3P2]  698: eva3 dw eva4-ev3
[KM0]  699: ev3:
[PX2]  700:      ;Schreiben Seite 3
[7M1]  701:    mov al,43h
[6H1]  702:    out dx,al
[PP2]  703: eva4 dw evae-ev4
[PM0]  704: ev4:
[TX2]  705:      ;Schreiben Seite 4
[9M1]  706:    mov al,44h
[6H1]  707:    out dx,al
[G31]  708: evae = $
[000]  709:
[6L5]  710:      ;Daten für den Header der EXE-Datei
[PD3]  711: exeheader dw exee-exe
[SD5]  712: exe dw "ZM",?,?,0,32,64,64,?,1024,0,0,0,30
[641]  713: exee = $
[000]  714:
[UY7]  715:      ;Die folgende Routine zeigt später das
       Bild an
[MK6]  716: vorspanndat dw vorspannende-differenz
[5S1]  717: differenz:
[UL0]  718:    cld
[BC1]  719:    push ds
[8C1]  720:    push cs
[E31]  721:    pop ds
[683]  722:    jmp short modusein
[VP7]  723:      ;Auf Tastendruck warten, wenn Option "c
       " fehlt
[321]  724: taste:
[E31]  725:    pop ds
[R64]  726:    cmp word ptr ds:[5dh]," C"
[1C2]  727:    je nichtwarten
[XK3]  728:      ;Cursor ausschalten
[7V1]  729:    mov cx,2000h
[LL1]  730:    mov ah,01h
[Y71]  731:    int 10h
[XU2]  732:      ;Auf Taste warten
[T91]  733:    mov ah,0
[G81]  734:    int 16h
[355]  735:      ;Videomodus wiederherstellen
[9B1]  736:    mov ax,0
[7Y3]  737: altermodus = byte ptr $-2
[Y71]  738:    int 10h
[XN2]  739:      ;Bild verlassen
[R52]  740: nichtwarten:
[VM1]  741:    mov ah,4ch
[381]  742:    int 21h
[2F6]  743:      ;Eine Seite (maximal 64 KByte) entpacke
       n
[US1]  744: entpacken:
```

Listing 1. »snapshot.asm« ist ein ausgereiftes und nützliches Hilfsprogramm für jeden Grafikfan

```
[EF1]  745:    xor di,di
[X91]  746:    mov ch,Ø
[FH1]  747: entpackØ:
[NLC]  748:      ;Quelldaten können unter widrigen Umstä
           nden größer als 64 KByte sein,
[ØF9]  749:      ;deshalb muß das Quellsegment bei Bedar
           f erhöht werden
[ET1]  75Ø:    cmp si,8ØØØh
[3V1]  751:    jb entpack1
[ET1]  752:    sub si,8ØØØh
[3H1]  753:    mov ax,ds
[GR1]  754:    add ax,Ø8ØØh
[1H1]  755:    mov ds,ax
[XK3]  756:      ;Neue Anweisung lesen
[GH1]  757: entpack1:
[CVØ]  758:    lodsb
[KF1]  759:    mov cl,al
[4H1]  76Ø:    cmp cl,128
[4Y4]  761:      ;Positiv: CX Werte übertragen
[7V1]  762:    jb entpack2
[EW2]  763:      ;128: Seite fertig
[MV1]  764:    je entpack3
[S48]  765:      ;Negativ: Folgender Wert wiederholt sic
           h - CX mal
[6HØ]  766: wØ:
[CVØ]  767:    lodsb
[HØ1]  768:    neg cl
[3PA]  769:      ;Rep-Präfix wegen des variablen Byte-/W
           ort-Zugriffs als Datum
[321]  77Ø:    db Øf3h
[9HØ]  771: w1:
[NXØ]  772:    stosb
[P42]  773:    jmp entpackØ
[5P2]  774:      ;Werte kopieren
[HH1]  775: entpack2:
[321]  776:    db Øf3h
[CHØ]  777: w2:
[PXØ]  778:    movsb
[P42]  779:    jmp entpackØ
[G52]  78Ø:      ;Seite fertig
[KH1]  781: entpack3:
[SN6]  782:      ;DS für nächste Seite normalisieren
[RF1]  783:    mov di,si
[DA1]  784:    mov cl,4
[ME1]  785:    shr di,cl
[3H1]  786:    mov ax,ds
[EC1]  787:    add ax,di
[1H1]  788:    mov ds,ax
[9E7]  789:      ;Sprung zur ersten Adresse hinter den D
           aten
[PC1]  79Ø:    push ax
[8B1]  791:    and si,15
[GC1]  792:    push si
[PA1]  793:      ;Ret-Far
[A31]  794:    db Øcbh
[WA3]  795:      ;Alten Modus retten
[7K1]  796: modusein:
[XM1]  797:    mov ah,Øfh
[Y71]  798:    int 1Øh
[FF5]  799:      ;Variablenzugriff zur Laufzeit!
[SN5]  8ØØ:    mov ds:[altermodus-differenz],al
[234]  8Ø1:      ;Neuen Modus einschalten
[9B1]  8Ø2:    mov ax,Ø
[GY3]  8Ø3: neuermodus = byte ptr $-2
[Y71]  8Ø4:    int 1Øh
[AH3]  8Ø5:      ;Zielsegment setzen
[9B1]  8Ø6:    mov ax,Ø
[LU3]  8Ø7: videobasis = word ptr $-2
[3H1]  8Ø8:    mov es,ax
[8C1]  8Ø9:    push cs
[1M2]  81Ø: vorspannende = $
[ØØØ]  811:
[4C8]  812:      ;Programmblock zwischen den einzelnen B
           ildseiten
[K83]  813: neueebene dw nee-neØ
[5LØ]  814: neØ:
[P94]  815:      ;SI auf Datenblock setzen
[M91]  816:    call ne1
[9LØ]  817: ne1:
[E31]  818:    pop si
[N32]  819:    add si,nee-ne1
[VM5]  82Ø:      ;Adresse des Entpackers auf Stack
[R31]  821:    pop ax
[PC1]  822:    push ax
[PC1]  823:    push ax
[2M4]  824:    mov ax,entpacken-differenz
[PC1]  825:    push ax
[W91]  826:      ;Ret Far
[A31]  827:    db Øcbh
[9Ø1]  828: nee = $
[ØØØ]  829:
[2D4]  83Ø:      ;Sprung zum Programmende
[9A3]  831: nachspann dw nse-ns
[XLØ]  832: ns:
[ZX3]  833:    mov ax,taste-differenz
[PC1]  834:    push ax
[A31]  835:    db Øcbh
[L11]  836: nse = $
```

```
[ØØØ]  837:
[ZP3]  838:      ;Flag für Wortzugriff
[BE1]  839: wort db Ø
[7A5]  84Ø:      ;Adressen der Byte-/Wort-Befehle
[A26]  841: worte dw wØ,w1,w2,w3,w4,w5,w6,w7,w8,w9,Ø
[4S1]  842: handle dw ?
[ØØØ]  843:
[KS4]  844:      ;Entpacker auf Disk schreiben
[HM1]  845: vorspann:
[XL6]  846:      ;Befehle an Byte-/Wortzugriff anpassen
[NE3]  847:    mov si,offset worte
[3B1]  848:    push di
[M42]  849: neuerbefehl:
[PWØ]  85Ø:    lodsw
[7E1]  851:    and ax,ax
[BS2]  852:      ;Adresse=Ø => Ende
[A22]  853:    je befehlende
[3G1]  854:    mov di,ax
[6G2]  855:    mov al,ds:[di]
[2S8]  856:      ;Bit Ø des Befehls ist für Wortzigriff
           verantwortlich
[XU1]  857:    and al,Øfeh
[WN1]  858:    or al,wort
[CH2]  859:    mov ds:[di],al
[NT2]  86Ø:    jmp neuerbefehl
[5Y1]  861: befehlende:
[G21]  862:    pop di
[UD4]  863:    mov si,offset vorspanndat
[ZN6]  864:      ;Datenblock [DS:SI] auf Disk schreiben
[XP1]  865: schreiben:
[WC1]  866:    push cx
[PWØ]  867:    lodsw
[EH1]  868:    mov cx,ax
[8D2]  869:    jcxz datenende
[722]  87Ø: neuesdatum:
[CVØ]  871:    lodsb
[8N2]  872:    call schreibeal
[LU2]  873:    loop neuesdatum
[ZR1]  874: datenende:
[V31]  875:    pop cx
[3PØ]  876:    ret
[ØØØ]  877:
[L9A]  878:      ;CX Bytes ab [VideoBasis:BX] gepackt au
           f Diskette schreiben
[P81]  879: packen:
[PK5]  88Ø:      ;Aufruf des Entpackers schreiben
[A24]  881:    mov si,offset neueebene
[ZC2]  882:    call schreiben
[TC1]  883:    push bx
[WC1]  884:    push cx
[ZC1]  885:    push dx
[BC1]  886:    push ds
[9Y2]  887:    mov ds,videobasis
[2H1]  888:    mov si,bx
[ED1]  889:    add cx,bx
[49A]  89Ø:      ;Endevergleich beim 1. Mal überspringen
           (für 64 KByte-Blöcke)
[UP3]  891:    jmp short ersterblock
[W12]  892: neuerblock:
[LL2]  893:      ;Ende erreicht?
[HF1]  894:    cmp si,cx
[BV1]  895:    je packende
[162]  896: ersterblock:
[GC1]  897:    push si
[SZ4]  898:      ;Bytezähler initialisieren
[BA1]  899:    mov dl,Ø
[FHØ]  9ØØ: w3:
[W63]  9Ø1:      ;Ersten Wert holen
[CVØ]  9Ø2:    lodsb
[CH1]  9Ø3:    mov bx,ax
[ØH3]  9Ø4:    jmp short pruefeende
[KHØ]  9Ø5: w4:
[2R3]  9Ø6:      ;Nächsten Wert holen
[CVØ]  9Ø7:    lodsb
[VR1]  9Ø8:    xchg ax,bx
[NHØ]  9Ø9: w5:
[7Ø5]  91Ø:      ;Mit vorigem Wert vergleichen
[YD1]  911:    cmp bl,al
[8C1]  912:    je gleich
[MØ1]  913:    inc dl
[5LC]  914:      ;Maximal 127 verschiedene Werte können
           an einem Stück verarbeitet werden
[3H1]  915:    cmp dl,127
[932]  916:    je schreibedl
[B12]  917: pruefeende:
[HF1]  918:    cmp si,cx
[WZØ]  919:    jne w4
[MØ1]  92Ø:    inc dl
[R96]  921:      ;Zähler und Werte auf Disk schreiben
[7Z1]  922: schreibedl:
[E31]  923:    pop si
[HF1]  924:    mov al,dl
[8N2]  925:    call schreibeal
[SHØ]  926: w6:
[CVØ]  927:    lodsb
[SP2]  928:    call schreibeax
[FZØ]  929:    dec dl
[2Ø1]  93Ø:    jne w6
```

```
[FH2]  931:    jmp neuerblock
[N81]  932: gleich:
[NL7]  933:      ;Falls Zähler > Ø, dann zuerst Werte sc
       hreiben
[T81]  934:    cmp dl,Ø
[HD2]  935:    jne schreibedl
[2G5]  936:      ;Zwei Werte sind bereits gleich
[CD1]  937:    mov dl,-2
[6B2]  938: pruefefertig:
[HF1]  939:    cmp si,cx
[TW1]  94Ø:    je ungleich
[VHØ]  941: w7:
[2R3]  942:      ;Nächsten Wert holen
[CVØ]  943:    lodsb
[YHØ]  944: w8:
[D54]  945:      ;Auf Gleichheit prüfen
[XD1]  946:    cmp al,bl
[732]  947:    jne ungleich
[FZØ]  948:    dec dl
[1DB]  949:      ;Maximal 127 gleiche Werte können an ei
       nem Stück verarbeitet werden
[AR1]  95Ø:    cmp dl,-127
[4X2]  951:    jne pruefefertig
[9H1]  952: ungleich:
[GY9]  953:      ;Zähler und sich wiederholender Wert au
       f Disk schreiben
[HF1]  954:    mov al,dl
[8N2]  955:    call schreibeal
[E31]  956:    pop si
[1KØ]  957: w9:
[CVØ]  958:    lodsb
[MØ1]  959:    inc dl
[BØ1]  96Ø:    jne w9
[SP2]  961:    call schreibeax
[FH2]  962:    jmp neuerblock
[RF1]  963: packende:
[FH4]  964:      ;Endekennzeichen schreiben
[KK1]  965:    mov al,128
[8N2]  966:    call schreibeal
[E31]  967:    pop ds
[X31]  968:    pop dx
[V31]  969:    pop cx
[T31]  97Ø:    pop bx
[3PØ]  971:    ret
[ØØØ]  972:
[KY8]  973:      ;Byte oder Wort in AX schreiben, je nac
       h Wert in "Wort"
[5Ø2]  974: schreibeax:
[4P2]  975:    cmp cs:[wort],Ø
[632]  976:    je schreibeal
[PC1]  977:    push ax
[8N2]  978:    call schreibeal
[R31]  979:    pop ax
[6F1]  98Ø:    mov al,ah
[FX3]  981:      ;AL in Puffer schreiben
[1Z1]  982: schreibeal:
[NXØ]  983:    stosb
[PA2]  984:      ;Puffer voll?
[YG1]  985:    cmp di,256
[5K2]  986:    jb schreibeende
[GK4]  987:      ;Puffer auf Disk schreiben
[YD2]  988: schreibeblock:
[5C1]  989:    and di,di
[EK2]  99Ø:    je schreibeende
[TC1]  991:    push bx
[WC1]  992:    push cx
[ZC1]  993:    push dx
[BC1]  994:    push ds
[GS2]  995:    mov bx,cs:handle
[LG1]  996:    mov cx,di
[8C1]  997:    push cs
[E31]  998:    pop ds
[TH1]  999:    xor dx,dx
[NL1] 1ØØØ:    mov ah,4Øh
[381] 1ØØ1:    int 21h
[E31] 1ØØ2:    pop ds
[X31] 1ØØ3:    pop dx
[V31] 1ØØ4:    pop cx
[T31] 1ØØ5:    pop bx
[DA6] 1ØØ6:      ;Bei Fehlern ist das Carryflag gesetzt
[R52] 1ØØ7:    jc schliessen
[C84] 1ØØ8:      ;Prüfen ob Diskette voll
[XE1] 1ØØ9:    cmp ax,di
[1G2] 1Ø1Ø:    jne schliessen
[TT4] 1Ø11:      ;Puferzeiger zurücksetzen
[EF1] 1Ø12:    xor di,di
[G62] 1Ø13: schreibeende:
[3PØ] 1Ø14:    ret
[R12] 1Ø15: schliessen:
[N62] 1Ø16:    mov bx,handle
[YM1] 1Ø17:    mov ah,3eh
[381] 1Ø18:    int 21h
[F91] 1Ø19: fehler:
[SA1] 1Ø2Ø:      ;Fiepen
[7X1] 1Ø21:    mov ax,Øe Ø7h
[Y71] 1Ø22:    int 1Øh
[W65] 1Ø23:      ;Rückkehr zum Hauptprogramm
[7H1] 1Ø24:    mov sp,bp
[3PØ] 1Ø25:    ret
[ØØØ] 1Ø26:
[3K2] 1Ø27: hauptprogramm:
[D97] 1Ø28:      ;Stackpointer für Fehlerrückkehr sicher
       n
[PG1] 1Ø29:    mov bp,sp
[5W2] 1Ø3Ø:    mov al,usermodus
[AV2] 1Ø31:    mov bh,userseite
[3E5] 1Ø32:      ;Moduserkennung auf Automatik?
[P74] 1Ø33:    cmp byte ptr automodus,Ø
[HN1] 1Ø34:    je manuell
[P86] 1Ø35:      ;Aktuellen Videomodus und Seite holen
[XM1] 1Ø36:    mov ah,Øfh
[Y71] 1Ø37:    int 1Øh
[2E1] 1Ø38: manuell:
[LV4] 1Ø39:      ;Videomodus in Tabelle suchen
[544] 1Ø4Ø:    mov si,offset videomodi
[952] 1Ø41: vergleichen:
[3F2] 1Ø42:    cmp al,ds:[si]
[RX1] 1Ø43:    je gefunden
[G23] 1Ø44:    add si,lengthvideo
[T14] 1Ø45:    cmp si,offset videoende
[5A2] 1Ø46:    jb vergleichen
[2T1] 1Ø47:    jmp fehler
[8G1] 1Ø48: gefunden:
[U13] 1Ø49:    mov neuermodus,al
[CD5] 1Ø5Ø:      ;Datei öffnen, Attribut: Archive
[3M1] 1Ø51:    mov cl,2Øh
[T23] 1Ø52:    mov dx,offset pfad
[UM1] 1Ø53:    mov ah,3ch
[381] 1Ø54:    int 21h
[P62] 1Ø55:    mov handle,ax
[X72] 1Ø56:    jnc oeffnenok
[1A1] 1Ø57: fehlerØ:
[2T1] 1Ø58:    jmp fehler
[1S1] 1Ø59: oeffnenok:
[8T8] 1Ø6Ø:      ;Pufferzeiger auf Ø setzen, Puffer ist
       nutzloser PSP
[EF1] 1Ø61:    xor di,di
[RC7] 1Ø62:      ;Unrelevante Werte im EXE-Kopf mit Ø fü
       llen
[AL1] 1Ø63:    mov cx,512
[Y82] 1Ø64: schreibenull:
[5A1] 1Ø65:    mov al,Ø
[8N2] 1Ø66:    call schreibeal
[313] 1Ø67:    loop schreibenull
[1Y3] 1Ø68:      ;Videomodus überlesen
[CVØ] 1Ø69:    lodsb
[H54] 1Ø7Ø:      ;Bildschirmsegment lesen
[PWØ] 1Ø71:    lodsw
[2Y2] 1Ø72:    mov videobasis,ax
[2E8] 1Ø73:      ;Seitenfaktor mit aktueller Seite multi
       plizieren
[CVØ] 1Ø74:    lodsb
[521] 1Ø75:    mul bh
[7A1] 1Ø76:    mov bl,Ø
[LH3] 1Ø77:      ;und mit 256 malnehmen
[4F1] 1Ø78:    mov bh,al
[TN3] 1Ø79:      ;Bytes pro Seite holen
[PWØ] 1Ø8Ø:    lodsw
[EH1] 1Ø81:    mov cx,ax
[DU4] 1Ø82:      ;Flag für Wortzugriff holen
[CVØ] 1Ø83:    lodsb
[142] 1Ø84:    mov wort,al
[X74] 1Ø85:      ;Unterprogramm aufrufen
[PWØ] 1Ø86:    lodsw
[891] 1Ø87:    call ax
[D44] 1Ø88:    mov si,offset nachspann
[ZC2] 1Ø89:    call schreiben
[H28] 1Ø9Ø:      ;Eventuelle Werte im Puffer auf Disk sc
       hreiben
[133] 1Ø91:    call schreibeblock
[HR4] 1Ø92:      ;Dateilänge nach DX:AX holen
[HV1] 1Ø93:    mov ax,42Ø2h
[N62] 1Ø94:    mov bx,handle
[PH1] 1Ø95:    xor cx,cx
[TH1] 1Ø96:    xor dx,dx
[381] 1Ø97:    int 21h
[EH1] 1Ø98:    jc fehlerØ
[DF6] 1Ø99:      ;Anzahl der Bytes in der letzten Seite
[PC1] 11ØØ:    push ax
[ØG1] 11Ø1:    and ax,511
[7A2] 11Ø2:    jne nichtvoll
[6L1] 11Ø3:    mov ax,512
[3U1] 11Ø4: nichtvoll:
[MØ2] 11Ø5:    mov exe+2,ax
[R31] 11Ø6:    pop ax
[AA6] 11Ø7:      ;Anzahl der 512-Byte-Seiten bestimmen
[PC1] 11Ø8:    push ax
[ZC1] 11Ø9:    push dx
[2F1] 111Ø:    add ax,511
[A71] 1111:    adc dx,Ø
[AL1] 1112:    mov cx,512
[931] 1113:    div cx
[SØ2] 1114:    mov exe+4,ax
[X31] 1115:    pop dx
[R31] 1116:    pop ax
```

Listing 1. »snapshot.asm« ist ein ausgereiftes und nützliches Hilfsprogramm für jeden Grafikfan

```
[E44] 1117:      ;Stacksegment berechnen
[KK1] 1118:    sub ax,497
[481] 1119:    sbb dx,Ø
[UE1] 112Ø:    mov cx,16
[931] 1121:    div cx
[F32] 1122:    mov exe+14,ax
[K65] 1123:      ;Dateizeiger auf Anfang setzen
[9V1] 1124:    mov ax,42ØØh
[PH1] 1125:    xor cx,cx
[TH1] 1126:    xor dx,dx
[381] 1127:    int 21h
[EH1] 1128:    jc fehlerØ
[ZK3] 1129:      ;Exe-Header schreiben
[S14] 113Ø:    mov si,offset exeheader
[ZC2] 1131:    call schreiben
[133] 1132:    call schreibeblock
[S13] 1133:      ;Datei schließen
[N62] 1134:    mov bx,handle
[YM1] 1135:    mov ah,3eh
[381] 1136:    int 21h
[ØW2] 1137:    jnc schliessenok
[2T1] 1138:    jmp fehler
[AA2] 1139: schliessenok:
[KL4] 114Ø:      ;Ende des Dateinamens suchen
[A23] 1141:    mov si,offset pfad
[LU1] 1142: suchenull:
[CVØ] 1143:    lodsb
[L81] 1144:    cmp al,Ø
[RA2] 1145:    jne suchenull
[KNØ] 1146:    std
[CVØ] 1147:    lodsb
[8V5] 1148:      ;Rückwärts nach Ziffern suchen
[LB2] 1149: pruefeziffer:
[SØ3] 115Ø:    cmp si,offset pfad
[V41] 1151:    jb ende
[CVØ] 1152:    lodsb
[P95] 1153:      ;Abbruch bei einem Subdirectory
[3H1] 1154:    cmp al,"\"
[451] 1155:    je ende
[BE1] 1156:    cmp al,"Ø"
[CP2] 1157:    jb pruefeziffer
[XE1] 1158:    cmp al,"9"
[9P2] 1159:    ja pruefeziffer
[BB5] 116Ø:      ;Gefundene Ziffer weiterzählen
[EØ1] 1161:    inc al
[FN2] 1162:    mov ds:[si+1],al
[XE1] 1163:    cmp al,"9"
[RA1] 1164:    jbe ende
[M78] 1165:      ;Bei Überlauf von "9" auf "Ø" nächste Z
                iffer suchen
[PC4] 1166:    mov byte ptr ds:[si+1],"Ø"
[UY2] 1167:    jmp pruefeziffer
[XTØ] 1168: ende:
[3PØ] 1169:    ret
[ØØØ] 117Ø:
[TM6] 1171:      ;Neue bzw. gerettete Interruptvektoren
[Ø84] 1172: alterintØ5 dw neuerintØ5,?
[K74] 1173: alterint1Ø dw neuerint1Ø,?
[P84] 1174: alterint19 dw neuerint19,?
[G84] 1175: alterintØ9 dw neuerintØ9,?
[C84] 1176: alterintØ8 dw neuerintØ8,?
[S84] 1177: alterint28 dw neuerint28,?
[884] 1178: alterint24 dw neuerint24,?
[X4A] 1179:      ;Zeiger auf Flag, das den Aufruf von DO
                S-Funktionen erlaubt
[5G2] 118Ø: indosflag dw ?,?
[UL5] 1181:      ;Verschiedene Schalterzustände
[X82] 1182: usermodus db Ø
[282] 1183: userseite db Ø
[FF2] 1184: automodus db -1
[Y72] 1185: hercmodus db Ø
[372] 1186: hercseite db Ø
[556] 1187:      ;Flag für ALT gedrückt (nur beim PC)
[HW1] 1188: altflag db Ø
[X54] 1189:      ;Flag für PrtSc verboten
[AH1] 119Ø: aktiv db Ø
[ØØØ] 1191:
[ØX2] 1192:      ;Hotkey erkennen
[PZ1] 1193: neuerintØ5:
[4X2] 1194:    assume ds:nothing
[ØØØ] 1195:
[2Ø2] 1196: neuerintØ9:
[PC1] 1197:    push ax
[M12] 1198: if intnr eq 9
[B56] 1199:      ;Scan-Cde der gedrückten Taste lesen
[H71] 12ØØ:    in al,6Øh
[2E3] 12Ø1:      ;Wurde Alt gedrückt?
[YK1] 12Ø2:    cmp al,38h
[ØG2] 12Ø3:    jne keinaltein
[BC2] 12Ø4:    mov altflag,-1
[K23] 12Ø5:    jmp short intØ9alt
[P12] 12Ø6: keinaltein:
[VZ3] 12Ø7:      ;Wurde Alt losgelassen?
[ZU1] 12Ø8:    cmp al,ØB8h
[3H2] 12Ø9:    jne keinaltaus
[Y72] 121Ø:    mov altflag,Ø
[K23] 1211:    jmp short intØ9alt
[T22] 1212: keinaltaus:
```

```
[SK1] 1213:    cmp al,54h
[Y82] 1214:    je sysrequest
[DF3] 1215:      ;Ist die Taste PrtSc?
[WK1] 1216:    cmp al,37h
[732] 1217:    jne intØ9alt
[886] 1218:      ;Ja: Ist gleichzeitig Alt gedrückt?
[C62] 1219:    cmp altflag,Ø
[VE6] 122Ø:      ;Nein, dann zur alten Tastaturroutine
[LV1] 1221:    je intØ9alt
[F72] 1222: sysrequest:
[TUØ] 1223: endif
[7B6] 1224:      ;Abbruch, wenn Programm bereits aktiv
[TX1] 1225:    cmp aktiv,Ø
[TG1] 1226:    jne adios
[GØ2] 1227:    mov aktiv,-1
[GC1] 1228:    push si
[3B1] 1229:    push di
[ØG6] 123Ø:      ;Ticker- und Dosfree-Interrupt setzen
[MG2] 1231:    call intØ8und28
[G21] 1232:    pop di
[E31] 1233:    pop si
[NØ1] 1234: adios:
[C12] 1235: if intnr eq 5
[R31] 1236:    pop ax
[GRØ] 1237:    iret
[ZRØ] 1238: else
[S35] 1239:      ;Empfangene Taste bestätigen
[K71] 124Ø:    in al,61h
[R61] 1241:    or al,128
[YN1] 1242:    out 61h,al
[CF1] 1243:    and al,127
[YN1] 1244:    out 61h,al
[696] 1245:      ;Interruptausführung bestätigen
[5M1] 1246:    mov al,61h
[LN1] 1247:    out 2Øh,al
[R31] 1248:    pop ax
[GRØ] 1249:    iret
[9H1] 125Ø: intØ9alt:
[R31] 1251:    pop ax
[V24] 1252:    jmp dword ptr alterintØ5
[TUØ] 1253: endif
[ØØØ] 1254:
[LT5] 1255:      ;Standardpalette für EGA und VGA
[4E7] 1256: egastandard db Ø,1,2,3,4,5,2Ø,7,56,57,58,59
            ,6Ø,61,62,63,Ø
[ØØØ] 1257:
[RP3] 1258:      ;Neuer Videointerrupt
[9Z1] 1259: neuerint1Ø:
[RB1] 126Ø:    and ah,ah
[ME2] 1261:    je modussetzen
[3L1] 1262:    cmp ah,Øbh
[WW1] 1263:    je cgafarben
[ØK1] 1264:    cmp ah,1Øh
[3C2] 1265:    jae egaodervga
[LK1] 1266: raushier:
[U24] 1267:    jmp dword ptr alterint1Ø
[3X4] 1268:      ;Neuen Videomodus einschalten
[872] 1269: modussetzen:
[UD7] 127Ø:      ;CGA- und EGA-Farben auf Standartwerte
                setzen
[482] 1271:    mov blinken,1
[WA2] 1272:    mov cgarand,16
[LP2] 1273:    mov cgapalette,Ø
[PC1] 1274:    push ax
[TC1] 1275:    push bx
[SE1] 1276:    mov bx,16
[5F2] 1277: naechstefarbe:
[Ø64] 1278:    mov al,[egastandard+bx]
[E14] 1279:    mov [egapalette+bx],al
[BZØ] 128Ø:    dec bl
[113] 1281:    jns naechstefarbe
[T31] 1282:    pop bx
[R31] 1283:    pop ax
[U52] 1284:    jmp raushier
[RP5] 1285:      ;CGA-Randfarbe bzw. Palette setzen
[MN1] 1286: cgafarben:
[981] 1287:    cmp bh,Ø
[C32] 1288:    jne keinrand
[ØE2] 1289:    mov cgarand,bl
[U52] 129Ø:    jmp raushier
[EH1] 1291: keinrand:
[XV2] 1292:    mov cgapalette,bl
[U52] 1293:    jmp raushier
[ØZ1] 1294: egaodervga:
[GT1] 1295:    cmp ax,1ØØØh
[T32] 1296:    jne overscan
[216] 1297:      ;Einzelnes Palettenregister setzen
[2C1] 1298:    cmp bl,16
[A32] 1299:    jae raushier
[3B1] 13ØØ:    push di
[4G1] 13Ø1:    mov di,bx
[AF1] 13Ø2:    and di,255
[VZ3] 13Ø3:    mov [egapalette+di],bh
[G21] 13Ø4:    pop di
[U52] 13Ø5:    jmp raushier
[VH1] 13Ø6: overscan:
[MT1] 13Ø7:    cmp ax,1ØØ1h
[882] 13Ø8:    jne farbblock
```

```
[3V2] 13Ø9:      ;Randfarbe setzen
[M83] 131Ø:    mov egapalette+16,bh
[U52] 1311:    jmp raushier
[4S1] 1312: farbblock:
[ST1] 1313:    cmp ax,1ØØ2h
[D92] 1314:    jne blinktest
[RG5] 1315:      ;Komplette Farbpalette neu setzen
[PC1] 1316:    push ax
[TC1] 1317:    push bx
[3B1] 1318:    push di
[6G1] 1319:    mov di,dx
[SE1] 132Ø:    mov bx,16
[NS1] 1321: neuefarbe:
[BW2] 1322:    mov al,es:[di+bx]
[E14] 1323:    mov [egapalette+bx],al
[BZØ] 1324:    dec bl
[HA2] 1325:    jns neuefarbe
[G21] 1326:    pop di
[T31] 1327:    pop bx
[R31] 1328:    pop ax
[U52] 1329:    jmp raushier
[9T1] 133Ø: blinktest:
[WT1] 1331:    cmp ax,1ØØ3h
[ØE1] 1332:    jne raus
[AG2] 1333:    mov blinken,bl
[2WØ] 1334: raus:
[U52] 1335:    jmp raushier
[ØØØ] 1336:
[HU6] 1337:      ;Interrupts bei Warmstart deaktivieren
[4Ø2] 1338: neuerint19:
[CG2] 1339:    call intØ5und19
[T81] 134Ø:    int 19h
[ØØØ] 1341:
[DF8] 1342:      ;Neuer Ticker-Interrupt bei aktiviertem
          Programm
[ZZ1] 1343: neuerintØ8:
[6P3] 1344:      ;Alte Routine bedienen
[ZXØ] 1345:    pushf
[144] 1346:    call dword ptr alterintØ8
[BC1] 1347:    push ds
[GC1] 1348:    push si
[GB4] 1349:      ;Sind DOS-Aufrufe erlaubt?
[7H4] 135Ø:    lds si,dword ptr indosflag
[DV3] 1351:    cmp byte ptr ds:[si],Ø
[E31] 1352:    pop si
[E31] 1353:    pop ds
[SP1] 1354:    jne geduld
[LD8] 1355:      ;Neuer Dosfree-Interrupt bei aktivierte
          m Programm
[3Ø2] 1356: neuerint28:
[PC1] 1357:    push ax
[TC1] 1358:    push bx
[WC1] 1359:    push cx
[ZC1] 136Ø:    push dx
[GC1] 1361:    push si
[3B1] 1362:    push di
[TB1] 1363:    push bp
[BC1] 1364:    push ds
[EC1] 1365:    push es
[SR7] 1366:      ;Ticker- und Dosfree-Interrupt restauri
          eren
[MG2] 1367:    call intØ8und28
[7T3] 1368:      ;Interrupts zulassen
[6PØ] 1369:    sti
[ØLB] 137Ø:      ;Fehlerbehandlungsroutine für Schreibfe
          hler setzen (erst ab DOS 3.Ø)
[CL1] 1371:    call int24
[2H1] 1372:    mov ax,cs
[3H1] 1373:    mov es,ax
[173] 1374:    call hauptprogramm
[695] 1375:      ;Fehlerbehandlung restaurieren
[CL1] 1376:    call int24
[G31] 1377:    pop es
[E31] 1378:    pop ds
[131] 1379:    pop bp
[G21] 138Ø:    pop di
[E31] 1381:    pop si
[X31] 1382:    pop dx
[V31] 1383:    pop cx
[T31] 1384:    pop bx
[R31] 1385:    pop ax
[KC4] 1386:      ;Neuen Aufruf ermöglichen
[BZ1] 1387:    mov aktiv,Ø
[E81] 1388: geduld:
[GRØ] 1389:    iret
[ØØØ] 139Ø:
[N84] 1391:      ;Schreibfehlerbehandlung
[RZ1] 1392: neuerint24:
[W75] 1393:      ;Code 3 entspricht (I)gnorieren
[8A1] 1394:    mov al,3
[GRØ] 1395:    iret
[ØØØ] 1396:
[NY5] 1397:      ;Hotkey-Erkennung ein-/ausschalten
[PY1] 1398: intØ5und19:
[C12] 1399: if intnr eq 5
[ØV1] 14ØØ:    mov di,Ø5h*4
[ZRØ] 14Ø1: else
[8V1] 14Ø2:    mov di,Ø9h*4
[TUØ] 14Ø3: endif
[V24] 14Ø4:    mov si,offset alterintØ5
[5H2] 14Ø5:    call interrupt
[RU1] 14Ø6:    mov di,1Øh*4
[U24] 14Ø7:    mov si,offset alterint1Ø
[5H2] 14Ø8:    call interrupt
[9V1] 14Ø9:    mov di,19h*4
[334] 141Ø:    mov si,offset alterint19
[WK3] 1411:    jmp short interrupt
[RN6] 1412:      ;Warten auf freies DOS ein-/ausschalten
[VY1] 1413: intØ8und28:
[BC1] 1414:    push ds
[EC1] 1415:    push es
[6V1] 1416:    mov di,Ø8h*4
[Y24] 1417:    mov si,offset alterintØ8
[5H2] 1418:    call interrupt
[8V1] 1419:    mov di,28h*4
[634] 142Ø:    mov si,offset alterint28
[5H2] 1421:    call interrupt
[G31] 1422:    pop es
[E31] 1423:    pop ds
[3PØ] 1424:    ret
[ZT5] 1425:      ;Fehlerbehandlung ein-/ausschalten
[MØ1] 1426: int24:
[ØV1] 1427:    mov di,24h*4
[234] 1428:    mov si,offset alterint24
[BC9] 1429:      ;Interruptvektor mit gesichertem Vektor
          vertauschen
[5X1] 143Ø: interrupt:
[ULØ] 1431:    cld
[2H1] 1432:    mov ax,cs
[1H1] 1433:    mov ds,ax
[GH1] 1434:    xor ax,ax
[3H1] 1435:    mov es,ax
[ZT2] 1436:    call vertauschen
[S72] 1437: vertauschen:
[BH2] 1438:    mov ax,es:[di]
[ØZØ] 1439:    movsw
[4P2] 144Ø:    mov ds:[si-2],ax
[3PØ] 1441:    ret
[ØØØ] 1442:
[2W7] 1443:      ;Dateiname beim ersten Aufruf ohne Name
          nsangabe
[GØ3] 1444: pfad db "BILDØ.EXE",Ø
[ØØØ] 1445:
[HA2] 1446: installieren:
[FA2] 1447:    assume ds:code
[7L4] 1448:    mov si,offset programmname
[KS1] 1449:    call write
[ØF9] 145Ø:      ;Segmentadresse der neuen Interruptproz
          eduren setzen
[994] 1451:    mov di,offset alterintØ5+2
[2H1] 1452:    mov ax,cs
[MB1] 1453:    mov cx,7
[6B2] 1454: setzesegment:
[ZYØ] 1455:    stosw
[361] 1456:    add di,2
[523] 1457:    loop setzesegment
[HA7] 1458:      ;Prüfen, ob Programm schon installiert
          ist
[XX2] 1459:    mov ax,35ØØh+intnr
[381] 146Ø:    int 21h
[C12] 1461: if intnr eq 5
[A44] 1462:    mov si,offset neuerintØ5
[ZRØ] 1463: else
[E44] 1464:    mov si,offset neuerintØ9
[TUØ] 1465: endif
[4G1] 1466:    mov di,bx
[ME1] 1467:    mov cx,32
[GT1] 1468:    repe cmpsw
[9L1] 1469:    je schonda
[ZX4] 147Ø:    mov si,offset installmeldung
[KS1] 1471:    call write
[S55] 1472:      ;Hotkey-Erkennung einschalten
[EMØ] 1473:    cli
[CG2] 1474:    call intØ5und19
[5Ø5] 1475:      ;Adresse des InDosFlags holen
[WL1] 1476:    mov ah,34h
[381] 1477:    int 21h
[NT2] 1478:    mov indosflag,bx
[923] 1479:    mov indosflag+2,es
[EP8] 148Ø:      ;Schaltervariablen liegen im aktuellen
          Code-Segment
[8C1] 1481:    push cs
[G31] 1482:    pop es
[665] 1483:      ;Dateinamen und Schalter lesen
[4C1] 1484: schonda:
[DL1] 1485:    mov si,128
[CVØ] 1486:    lodsb
[8W5] 1487:      ;Parameterstring mit Ø abschließen
[T91] 1488:    mov ah,Ø
[ØD1] 1489:    add si,ax
[ZW3] 149Ø:    mov byte ptr ds:[si],Ø
[GL1] 1491:    mov si,129
[Ø24] 1492:      ;Leerzeichen überlesen
[842] 1493: leerzeichen:
[CVØ] 1494:    lodsb
```

Listing 1. »snapshot.asm« ist ein ausgereiftes und nützliches Hilfsprogramm für jeden Grafikfan

```
[L81] 1495:   cmp al,Ø
[422] 1496:   je keinparamØ
[BD1] 1497:   cmp al," "
[RA2] 1498:   je leerzeichen
[9E1] 1499:   cmp al,"/"
[3X1] 15ØØ:   je schalter
[PZ3] 15Ø1:     ;Dateinamen übertragen
[C13] 15Ø2:   mov di,offset pfad
[AG6] 15Ø3:     ;Maximal 63 Zeichen mit Null-Abschluß
[7D1] 15Ø4:   mov ah,64
[Y52] 15Ø5: uebertragen:
[XYØ] 15Ø6:   dec ah
[BR1] 15Ø7:   jne pfadok
[3G4] 15Ø8:   mov si,offset laengefehler
[KS1] 15Ø9:   call write
[NY1] 151Ø: keinparamØ:
[Ø23] 1511:   jmp keinparameter
[191] 1512: pfadok:
[6P7] 1513:     ;Der Name wird mit Ø, Leerzeichen oder
          "/" beendet
[L81] 1514:   cmp al,Ø
[422] 1515:   je keinparamØ
[9E1] 1516:   cmp al,"/"
[3X1] 1517:   je schalter
[BD1] 1518:   cmp al," "
[AU2] 1519:   je neuerschalterØ
[ED7] 152Ø:     ;Kleinbuchstaben in Großbuchstaben wand
          eln
[DF1] 1521:   cmp al,"a"
[AM2] 1522:   jb nichtwandeln
[ZG1] 1523:   cmp al,"z"
[7M2] 1524:   ja nichtwandeln
[ST1] 1525:   and al,255-32
[L82] 1526: nichtwandeln:
[NXØ] 1527:   stosb
[M74] 1528:     ;Namen mit Ø abschließen
[NV3] 1529:   mov byte ptr es:[di],Ø
[2L4] 153Ø:     ;Nächstes Zeichen prüfen
[CVØ] 1531:   lodsb
[ZU2] 1532:   jmp uebertragen
[YH1] 1533: schalter:
[CVØ] 1534:   lodsb
[R56] 1535:     ;Schalter in Kleinbuchstaben wandeln
[F31] 1536:   or al,32
[DF1] 1537:   cmp al,"a"
[N13] 1538:   jne keinautomatik
[DP3] 1539:   mov es:[automodus],-1
[CS2] 154Ø: neuerschalterØ:
[F33] 1541:   jmp neuerschalter
[HH2] 1542: keinautomatik:
[VF1] 1543:   cmp al,"h"
[5Ø3] 1544:   jne keinhercmodus
[RM3] 1545:   mov es:[hercmodus],-1
[ØS5] 1546:     ;Ist eine Seite bei "/h" angegeben?
[CVØ] 1547:   lodsb
[3E1] 1548:   cmp al,","
[113] 1549:   jne sucheschalter
[CVØ] 155Ø:   lodsb
[LY4] 1551:     ;Mögliche Seiten sind Ø und 1
[BE1] 1552:   cmp al,"Ø"
[3C2] 1553:   je hercseiteok
[DE1] 1554:   cmp al,"1"
[3C2] 1555:   je hercseiteok
[TØ3] 1556:   jmp falscheoption
[M52] 1557: hercseiteok:
[BE1] 1558:   sub al,"Ø"
[6P3] 1559:   mov es:[hercseite],al
[D74] 156Ø:   jmp short neuerschalter
[SP1] 1561: schalterØ:
[552] 1562:   jmp schalter
[1G2] 1563: keinhercmodus:
[5G1] 1564:   cmp al,"m"
[HU2] 1565:   jne keinmdamodus
[SH3] 1566:   mov es:[hercmodus],Ø
[D74] 1567:   jmp short neuerschalter
[L82] 1568: keinmdamodus:
[GC8] 1569:     ;Ist der Schalter eine gültige Hexadezi
          malziffer?
[PD2] 157Ø:   call hexziffer
[GV2] 1571:   jc falscheoptionØ
[DA1] 1572:   mov cl,4
[ZD1] 1573:   shl al,cl
[PC1] 1574:   push ax
[L64] 1575:     ;Zweite Hexziffer holen
[CVØ] 1576:   lodsb
[F31] 1577:   or al,32
[PD2] 1578:   call hexziffer
[V31] 1579:   pop cx
[GV2] 158Ø:   jc falscheoptionØ
[961] 1581:   or al,cl
[5P5] 1582:     ;Ist der angegebene Modus bekannt?
[G44] 1583:   mov bx,offset videomodi
[H11] 1584: suche:
[HF2] 1585:   cmp al,es:[bx]
[RP2] 1586:   je modusbekannt
[V23] 1587:   add bx,lengthvideo
[424] 1588:   cmp bx,offset videoende
[581] 1589:   jb suche
```

```
[DC4] 159Ø:   mov si,offset modusfehler
[P44] 1591:   jmp short fehlerausgabe
[392] 1592: modusbekannt:
[ØS3] 1593:   mov es:[usermodus],al
[EL3] 1594:   mov es:[automodus],Ø
[3S6] 1595:     ;Ist außerdem noch eine Seite angegeben
          ?
[CVØ] 1596:   lodsb
[3E1] 1597:   cmp al,","
[113] 1598:   jne sucheschalter
[CVØ] 1599:   lodsb
[PD2] 16ØØ:   call hexziffer
[PN2] 16Ø1: falscheoptionØ:
[GS2] 16Ø2:   jc falscheoption
[5R3] 16Ø3:   mov es:[userseite],al
[U26] 16Ø4:     ;Folgt eine weitere Schalterangabe?
[2K2] 16Ø5: neuerschalter:
[CVØ] 16Ø6:   lodsb
[XG2] 16Ø7: sucheschalter:
[L81] 16Ø8:   cmp al,Ø
[GP2] 16Ø9:   je keinparameter
[BD1] 161Ø:   cmp al," "
[AR2] 1611:   je neuerschalter
[9E1] 1612:   cmp al,"/"
[LY1] 1613:   je schalterØ
[PK4] 1614:   mov si,offset syntaxfehler
[TC2] 1615: fehlerausgabe:
[KS1] 1616:   call write
[LG2] 1617: keinparameter:
[NØ5] 1618:     ;Einschaltmeldungen ausgeben
[6PØ] 1619:   sti
[RA4] 162Ø:   mov si,offset pfadmeldung
[KS1] 1621:   call write
[BC1] 1622:   push ds
[EC1] 1623:   push es
[E31] 1624:   pop ds
[A23] 1625:   mov si,offset pfad
[KS1] 1626:   call write
[E31] 1627:   pop ds
[XK4] 1628:   mov si,offset modusmeldung
[KS1] 1629:   call write
[WH3] 163Ø:   cmp es:[automodus],Ø
[YA2] 1631:   jne automatik
[AU3] 1632:   mov dl,es:[usermodus]
[ZC1] 1633:   push dx
[DA1] 1634:   mov cl,4
[XE1] 1635:   shr dl,cl
[792] 1636:   call writehex
[X31] 1637:   pop dx
[LA1] 1638:   and dl,15
[792] 1639:   call writehex
[UH4] 164Ø:   mov si,offset seitemeldung
[KS1] 1641:   call write
[TT3] 1642:   mov dl,es:[userseite]
[792] 1643:   call writehex
[RD2] 1644:   jmp short crlf
[CF2] 1645: falscheoption:
[LW4] 1646:   mov si,offset schalterfehler
[6Y2] 1647:   jmp fehlerausgabe
[UU1] 1648: automatik:
[HB4] 1649:   mov si,offset automeldung
[KS1] 165Ø:   call write
[6VØ] 1651: crlf:
[DS3] 1652:   mov si,offset linefeed
[KS1] 1653:   call write
[6G3] 1654:   cmp es:[hercmodus],Ø
[4U2] 1655:   je keinhercgrafik
[8F4] 1656:   mov si,offset hercmeldung1
[KS1] 1657:   call write
[VR3] 1658:   mov dl,es:[hercseite]
[792] 1659:   call writehex
[BF4] 166Ø:   mov si,offset hercmeldung2
[KS1] 1661:   call write
[AR2] 1662: keinhercgrafik:
[C54] 1663:     ;Ist es ein Erstaufruf?
[2H1] 1664:   mov ax,cs
[6H1] 1665:   mov bx,es
[TF1] 1666:   cmp ax,bx
[142] 1667:   jne zurueck
[SA5] 1668:     ;Ja, dann resident installieren
[3D3] 1669:   mov dx,offset pfad+64
[N81] 167Ø:   int 27h
[9G1] 1671: zurueck:
[VM1] 1672:   mov ah,4ch
[381] 1673:   int 21h
[ØØØ] 1674:
[BU7] 1675:     ;Den String [DS:SI] ausgeben, abschluß
          mit Ø
[BG1] 1676: writeit:
[MF1] 1677:   mov dl,al
[XL1] 1678:   mov ah,Ø6h
[381] 1679:   int 21h
[M21] 168Ø: write:
[CVØ] 1681:   lodsb
[BC1] 1682:   and al,al
[342] 1683:   jne writeit
[3PØ] 1684:   ret
[ØØØ] 1685:
```

```
[Y29] 1686:      ;Das ASCII-Zeichen in AL auf gültige He
          xziffer prüfen
[CT1] 1687: hexziffer:
[BE1] 1688:   sub al,"Ø"
[EZ1] 1689:   jc hexfehler
[MB1] 169Ø:   cmp al,1Ø
[DL1] 1691:   jb hexende
[7C1] 1692:   cmp al,48
[E82] 1693:   jbe hexfehler
[8C1] 1694:   sub al,39
[YB1] 1695:   cmp al,15
[8Z1] 1696:   ja hexfehler
[UD1] 1697: hexende:
[PLØ] 1698:   clc
[3PØ] 1699:   ret
[ET1] 17ØØ: hexfehler:
[ENØ] 17Ø1:   stc
[3PØ] 17Ø2:   ret
[ØØØ] 17Ø3:
[PL5] 17Ø4:     ;Eine hexadezimale Ziffer ausgeben
[YK1] 17Ø5: writehex:
[LA1] 17Ø6:   and dl,15
[2C1] 17Ø7:   add dl,"Ø"
[3F1] 17Ø8:   cmp dl,"9"
[EX2] 17Ø9:   jbe keinbuchstabe
[H61] 171Ø:   add dl,7
[EE2] 1711: keinbuchstabe:
[NL1] 1712:   mov ah,Ø2h
[381] 1713:   int 21h
[3PØ] 1714:   ret
[ØØØ] 1715:
[3Y6] 1716: programmname db 1Ø,"SnapShot Version 2.7",Ø
[DH7] 1717: installmeldung db " installiert. Aufruf mit
          "
[C12] 1718: if intnr eq 5
[532] 1719:   db "<PrtSc>",Ø
[ZRØ] 172Ø: else
[M45] 1721:   db "<Alt-PrtSc> bzw. <SysReq>",Ø
[TUØ] 1722: endif
[H17] 1723: laengefehler db 13,1Ø,"Fehler: Der angegebe
          ne "
[3Y5] 1724:   db "Pfad ist länger als 62 Zeichen",Ø
[WØ9] 1725: schalterfehler db 13,1Ø,"Fehler: Erlaubte O
          ptionen sind "
[7V6] 1726:   db "/Modus[,Seite], /A, /H[,Seite] und /M
          ",Ø
[MØ7] 1727: modusfehler db 13,1Ø,"Fehler: Der angegeben
          e "
[1L5] 1728:   db "Videomodus ist nicht bekannt",Ø
[146] 1729: syntaxfehler db 13,1Ø,"Fehler: Syntax "
[4L6] 173Ø:   db "ist Snapshot [Pfad] [/Optionen]",Ø
[887] 1731: pfadmeldung db 13,1Ø,"  Aktueller Dateiname
          : ",Ø
[HH7] 1732: modusmeldung db 13,1Ø,"  Eingestellter Modu
          s: ",Ø
[L84] 1733: seitemeldung db ", Seite ",Ø
[FY6] 1734: automeldung db "Automatische Erkennung",Ø
[6X8] 1735: hercmeldung1 db "  Im Modus Ø7 wird die Her
          culesgrafik "
[MC2] 1736:   db "auf Seite ",Ø
[7X4] 1737: hercmeldung2 db " gespeichert"
[RU2] 1738: linefeed db 13,1Ø,Ø
[ØØØ] 1739:
[TF1] 174Ø: code ends
[SF1] 1741: end begin
```

Listing 1. »snapshot.asm« ist ein ausgereiftes und nützliches Hilfsprogramm für jeden Grafikfan

Name	snapshot.com
Sprache	Assembler
Grafikkarte	jede
Besonderheiten	Bereits übersetzte Version von »snapshot.asm« (=Hexdump). Bitte mit dem MS-DOS-Debugger oder einer Textverarbeitung und SCheck eingeben.

```
[X95]     1: E9 93 Ø9 ØØ ØØ B8 Ø8 DØ Ø7 Ø1 Ø8 Ø2 Ø1 ØØ B8 Ø8
[375]     2: DØ Ø7 Ø1 Ø8 Ø2 Ø2 ØØ B8 1Ø AØ ØF Ø1 Ø8 Ø2 Ø3 ØØ
[RB5]     3: B8 1Ø AØ ØF Ø1 Ø8 Ø2 Ø4 ØØ B8 4Ø 4Ø 3F ØØ 23 Ø2
```

```
[A95]     4: Ø5 ØØ B8 4Ø 4Ø 3F ØØ 23 Ø2 Ø6 ØØ B8 4Ø 4Ø 3F ØØ
[9D5]     5: 3F Ø2 Ø7 ØØ BØ 1Ø AØ ØF Ø1 CD Ø2 ØD ØØ AØ 2Ø 4Ø
[TF5]     6: 1F ØØ 7D Ø3 ØE ØØ AØ 4Ø 8Ø 3E ØØ 7D Ø3 ØF ØØ AØ
[LA5]     7: 8Ø 6Ø 6D ØØ 7D Ø3 1Ø ØØ AØ 8Ø 6Ø 6D ØØ 7D Ø3 11
[V75]     8: ØØ AØ ØØ ØØ 96 ØØ 1D Ø2 12 ØØ AØ ØØ ØØ 96 ØØ 7D
[2C5]     9: Ø3 13 ØØ AØ ØØ ØØ FA ØØ 38 Ø4 7C ØØ B8 ØØ ØØ 7D
[E95]    1Ø: Ø1 3A Ø5 18 ØØ BØ 4Ø 6Ø 2D Ø1 1D Ø2 19 ØØ BØ 1Ø
[2B5]    11: C8 19 Ø1 1D Ø2 1A ØØ BØ 1Ø EØ 1C Ø1 1D Ø2 22 ØØ
[GC5]    12: B8 4Ø 6Ø 2D Ø1 1D Ø2 23 ØØ B8 2Ø C8 19 Ø1 1D Ø2
[UA5]    13: 24 ØØ B8 2Ø EØ 1C Ø1 1D Ø2 25 ØØ AØ ØØ ØØ 96 ØØ
[L95]    14: 7D Ø3 26 ØØ B8 4Ø 8Ø 25 Ø1 1D Ø2 29 ØØ AØ ØØ 6Ø
[KF5]    15: EA ØØ 7D Ø3 2A ØØ B8 2Ø 4Ø 1F Ø1 1D Ø2 2D ØØ AØ
[K75]    16: ØØ ØØ ØØ ØØ BE Ø5 2E ØØ AØ ØØ ØØ ØØ ØØ BE Ø5 37
[GD5]    17: ØØ AØ ØØ ØØ ØØ ØØ 7D Ø3 E8 ED Ø4 BE 14 Ø2 E8 Ø2
[2F5]    18: Ø5 E9 ØD Ø5 Ø7 ØØ B8 Ø3 1Ø B3 Ø1 CD 1Ø E8 D8 Ø4
[6S5]    19: E9 FE Ø4 E8 D2 Ø4 BE 2F Ø2 E8 E7 Ø4 E9 F2 Ø4 ØE
[UF5]    2Ø: ØØ B4 ØB BB 1Ø ØØ CD 1Ø B4 ØB BB ØØ Ø1 CD 1Ø 53
[7S5]    21: 33 DB B4 Ø6 CD 15 23 DB 5B 74 D2 53 51 1E 8E 1E
[1X5]    22: C1 Ø6 BA DE Ø3 33 F6 B9 AØ 1F BØ ØØ EE 8B 1C FE
[DW5]    23: CØ EE 39 1C 75 ØF FE CØ EE 39 1C 75 Ø8 FE CØ EE
[LU5]    24: AD 3B C3 E1 E5 1F 59 5B 74 A3 E8 7B Ø4 BE BØ Ø2
[5T5]    25: E8 9Ø Ø4 BA DE Ø3 BØ ØØ EE E8 95 Ø4 BE BE Ø2 E8
[ØL5]    26: 81 Ø4 BØ Ø1 EE E8 89 Ø4 BE C3 Ø2 E8 75 Ø4 BØ Ø2
[NP5]    27: EE E8 7D Ø4 BE C8 Ø2 E8 69 Ø4 BØ Ø3 EE E9 71 Ø4
[5K5]    28: ØC ØØ BA D9 Ø3 BØ ØF EE 83 C2 Ø4 BØ Ø1 EE Ø3 ØØ
[BL5]    29: BØ Ø2 EE Ø3 ØØ BØ Ø4 EE Ø3 ØØ BØ Ø8 EE 8Ø 3E FØ
[9K5]    3Ø: Ø8 ØØ 75 58 C6 Ø6 F1 Ø8 ØØ 1E B8 4Ø ØØ 8E D8 AØ
[TC5]    31: 65 ØØ 1F A8 Ø2 74 ØB A8 8Ø 74 41 C6 Ø6 F1 Ø8 Ø1
[MV5]    32: EB 3A BA B9 Ø3 EE 42 EC A8 Ø2 74 Ø8 42 EE 4A EC
[EU5]    33: A8 Ø2 74 Ø3 E9 16 FF EC A8 8Ø 74 FB EC A8 8Ø 75
[GY5]    34: FB 42 EE 83 EA Ø2 EE BA B4 Ø3 BØ 1Ø EE 42 EC 8A
[ZU5]    35: EØ 4A BØ 11 EE 42 EC 3D 8E ØB 72 D8 C6 Ø6 DF Ø6
[VN5]    36: ØØ E8 C4 Ø3 BE 49 Ø3 E8 D9 Ø3 AØ F1 Ø8 B4 8Ø F6
[FW5]    37: E4 8A F8 B9 EE 7E E9 D8 Ø3 32 ØØ BA BF Ø3 BØ Ø1
[HB5]    38: EE E8 18 ØØ ØØ 35 Ø1 2D Ø2 2E Ø3 Ø7 Ø4 5B Ø5 Ø2
[D95]    39: Ø6 57 Ø7 57 Ø8 Ø2 Ø9 Ø3 ØA ØØ ØB ØØ 5E B9 ØC ØØ
[DY5]    4Ø: BA B4 Ø3 AD EF E2 FC BA B8 Ø3 BØ ØA EE E8 78 Ø3
[ØN5]    41: 8Ø 3E BC Ø6 ØF 74 Ø6 BE F6 Ø3 E8 86 Ø3 BE 16 Ø4
[YP5]    42: E8 8Ø Ø3 BA CE Ø3 BØ Ø4 EE 42 BØ ØØ EE E8 81 Ø3
[5P5]    43: 8Ø 3E BC Ø6 ØF 74 ØC BE 22 Ø4 E8 66 Ø3 BØ Ø1 EE
[MH5]    44: E8 6E Ø3 8Ø 3E BC Ø6 29 74 1F BE 27 Ø4 E8 53 Ø3
[RP5]    45: BØ Ø2 EE E8 5B Ø3 8Ø 3E BC Ø6 ØF 74 ØC BE 2C Ø4
[3M5]    46: E8 4Ø Ø3 BØ Ø3 EE E8 48 Ø3 23 C9 75 18 8Ø 3E BC
[3D5]    47: Ø6 37 75 11 C7 Ø6 C1 Ø6 ØØ BØ B9 ØØ 8Ø BE 31 Ø4
[FA5]    48: E8 2Ø Ø3 EB 98 C3 1E ØØ E8 11 ØØ ØØ ØØ ØØ ØØ ØØ
[U25]    49: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 5A Ø6 ØE Ø7
[6G5]    5Ø: B8 Ø2 1Ø CD 1Ø Ø7 ØA ØØ BA C4 Ø3 BØ Ø2 EE 42 BØ
[WE5]    51: Ø1 EE Ø3 ØØ BØ Ø2 EE Ø3 ØØ BØ Ø4 EE Ø3 ØØ BØ Ø8
[7K5]    52: EE Ø5 ØØ B8 ØØ BØ 8E CØ E8 BD Ø2 E8 43 Ø1 1E 33
[YM5]    53: CØ 8E D8 8Ø 3E 84 Ø4 18 1F 75 Ø3 E9 D3 Ø2 B9 CØ
[9P5]    54: A8 BA CE Ø3 BØ Ø4 EE 42 BE 88 Ø4 E8 B5 Ø2 BØ Ø2
[FS5]    55: EE E8 BD Ø2 BE 24 Ø5 E8 A9 Ø2 BØ Ø1 EE E8 B1 Ø2
[3S5]    56: BE 3Ø Ø5 E8 9D Ø2 BØ Ø3 EE E8 A5 Ø2 BE 35 Ø5 E8
[LL5]    57: 91 Ø2 BØ ØØ EE E9 99 Ø2 9A ØØ E8 3B ØØ 2D ØØ 3B
[XA5]    58: Ø8 ØØ 3Ø 75 Ø1 Ø1 ØF ØØ Ø6 Ø3 ØØ 6D 59 5A 8E 62
[VC5]    59: 8B ØB 3E ØØ 4Ø ØØ ØØ ØØ ØØ ØØ ØØ EA 8C DF 2D ØØ
[PP5]    6Ø: E7 Ø4 C3 FF ØØ ØØ ØØ ØØ ØØ 4Ø Ø5 ØF FF 86 C4 EE
[CX5]    61: 42 86 C4 EE 4A FE C4 C3 5E Ø6 57 33 CØ 8E CØ BF
[5Y5]    62: 4A Ø4 A5 BF 84 Ø4 A4 A5 BF 4C Ø4 A5 5F Ø7 FA BA
[WS5]    63: C4 Ø3 B4 ØØ AC E8 D5 FF 8Ø FC Ø5 72 F7 B2 C2 BØ
[ZZ5]    64: C7 EE B2 C4 AD E8 C5 FF FB B2 D4 B8 11 ØØ EE 42
[N16]    65: EC 24 7F EE 4A AC E8 B4 FF 8Ø FC 19 72 F7 B2 CC
[1W5]    66: BØ ØØ EE B2 CA B8 Ø1 ØØ EE B2 CE AC E8 9E FF 8Ø
[ES5]    67: FC Ø9 72 F7 ØA ØØ BA C4 Ø3 BØ Ø2 EE 42 BØ Ø3 EE
[9H5]    68: Ø3 ØØ BØ Ø8 EE Ø3 ØØ BØ Ø1 EE C6 Ø6 BC Ø6 Ø3 9Ø
[3T5]    69: E8 B5 Ø1 BE 4C Ø5 E8 CA Ø1 E9 D5 Ø1 33 ØØ E8 ØB
[GE5]    7Ø: ØØ 71 5Ø 5A ØA 7F Ø6 64 7Ø ØØ Ø1 2Ø 5E BA D4 Ø3
[KX5]    71: 32 E4 8A C4 EE 42 AC EE 4A FE C4 8Ø FC ØB 72 F2
[5N5]    72: B2 D8 BØ Ø9 EE Ø6 33 CØ 8E CØ 26 C6 Ø6 49 Ø4 7C
[CH5]    73: Ø7 BE A8 Ø5 E8 8C Ø1 51 BA C7 Ø3 32 CØ EE 42 42
[SS5]    74: B9 ØØ Ø1 EC E8 ØD Ø2 EC E8 Ø9 Ø2 EC E8 Ø5 Ø2 E2
[UK5]    75: F2 59 BE AD Ø5 E9 6B Ø1 Ø3 ØØ E8 ØØ Ø3 ØF ØØ 5A
[FL5]    76: Ø6 ØE Ø7 B8 12 1Ø 33 DB B9 ØØ Ø1 CD 1Ø Ø7 E8 37
[GX5]    77: Ø1 E8 BD FF BE 1B Ø6 E8 49 Ø1 BA CD Ø3 BØ 4Ø EE
[KT5]    78: E8 4E Ø1 BE 23 Ø6 E8 3A Ø1 BØ 48 EE E8 42 Ø1 BE
[CS5]    79: 28 Ø6 E8 2E Ø1 BØ 5Ø EE E8 36 Ø1 BE 2D Ø6 E8 22
[KM5]    8Ø: Ø1 BØ 58 EE 8Ø 3E BC Ø6 2D 75 Ø6 B9 ØØ 6B E9 2Ø
[YL5]    81: Ø1 E8 1D Ø1 BE 32 Ø6 E8 Ø9 Ø1 BØ 6Ø EE 8Ø 3E BC
[ZK5]    82: Ø6 2E 75 Ø6 B9 ØØ BØ E9 Ø7 Ø1 C3 Ø6 ØØ BA CD Ø3
[XH5]    83: BØ 4Ø EE Ø3 ØØ BØ 41 EE Ø3 ØØ BØ 42 EE Ø3 ØØ BØ
[4B5]    84: 43 EE Ø3 ØØ BØ 44 EE 1A ØØ 4D 5A ØØ ØØ ØØ ØØ ØØ
[PZ4]    85: ØØ 2Ø ØØ 4Ø ØØ 4Ø ØØ ØØ ØØ ØØ Ø4 ØØ ØØ ØØ ØØ ØØ
[AH5]    86: ØØ 1E ØØ 71 ØØ FC 1E ØE 1F EB 59 1F 81 3E 5D ØØ
[6E5]    87: 43 2Ø 74 1Ø B9 ØØ 2Ø B4 Ø1 CD 1Ø B4 ØØ CD 16 B8
[GG5]    88: ØØ ØØ CD 1Ø B4 4C CD 21 33 FF B5 ØØ 81 FE ØØ 8Ø
[7R5]    89: 72 ØB 81 EE ØØ 8Ø 8C D8 Ø5 ØØ Ø8 8E D8 AC 8A C8
[2W5]    9Ø: 8Ø F9 8Ø 72 Ø9 74 ØB AC F6 D9 F3 AA EB DE F3 A4
[7Y5]    91: EB DA 8B FE B1 Ø4 D3 EF 8C D8 Ø3 C7 8E D8 5Ø 83
[8N5]    92: E6 ØF 56 CB B4 ØF CD 1Ø A2 1B ØØ B8 ØØ ØØ CD 1Ø
[7E5]    93: B8 ØØ ØØ 8E CØ ØE 1Ø ØØ E8 ØØ ØØ 5E 81 C6 ØD ØØ
[DF5]    94: 58 5Ø 5Ø B8 23 ØØ 5Ø CB Ø5 ØØ B8 Ø6 ØØ 5Ø CB ØØ
[TE5]    95: 97 Ø6 9B Ø6 9F Ø6 3C Ø7 41 Ø7 43 Ø7 5A Ø7 6F Ø7
```

Listing 2. Wenn Sie keinen Assembler besitzen, dann tippen Sie das Hexdump »snapshot.com« ab

```
[TD5]    96: 7Ø Ø7 81 Ø7 ØØ ØØ ØØ ØØ BE EØ Ø6 57 AD 23 CØ 74
[SV5]    97: ØE 8B F8 8A Ø5 24 FE ØA Ø6 DF Ø6 88 Ø5 EB ED 5F
[YS5]    98: BE 53 Ø6 51 AD 8B C8 E3 Ø6 AC E8 87 ØØ E2 FA 59
[KS5]    99: C3 BE C6 Ø6 E8 EC FF 53 51 52 1E 8E 1E C1 Ø6 8B
[2W5]   1ØØ: F3 Ø3 CB EB Ø4 3B F1 74 52 56 B2 ØØ AC 8B D8 EB
[SU5]   1Ø1: ØD AC 93 3A D8 74 1D FE C2 8Ø FA 7F 74 Ø6 3B F1
[2Ø6]   1Ø2: 75 EF FE C2 5E 8A C2 E8 4A ØØ AC E8 37 ØØ FE CA
[YT5]   1Ø3: 75 F8 EB D1 8Ø FA ØØ 75 EB B2 FE 3B F1 74 ØC AC
[AR5]   1Ø4: 3A C3 75 Ø7 FE CA 8Ø FA 81 75 FØ 8A C2 E8 24 ØØ
[ES5]   1Ø5: 5E AC FE C2 75 FB E8 ØC ØØ EB AA BØ 8Ø E8 14 ØØ
[MN5]   1Ø6: 1F 5A 59 5B C3 2E 8Ø 3E DF Ø6 ØØ 74 Ø7 5Ø E8 Ø3
[TG5]   1Ø7: ØØ 58 8A C4 AA 81 FF ØØ Ø1 72 23 23 FF 74 1F 53
[4P5]   1Ø8: 51 52 1E 2E 8B 1E F6 Ø6 8B CF ØE 1F 33 D2 B4 4Ø
[5P5]   1Ø9: CD 21 1F 5A 59 5B 72 Ø7 3B C7 75 Ø3 33 FF C3 8B
[4T5]   11Ø: 1E F6 Ø6 B4 3E CD 21 B8 Ø7 ØE CD 1Ø 8B E5 C3 8B
[NR5]   111: EC AØ ED Ø8 8A 3E EE Ø8 8Ø 3E EF Ø8 ØØ 74 Ø4 B4
[7M5]   112: ØF CD 1Ø BE Ø3 Ø1 3A Ø4 74 ØB 83 C6 Ø9 81 FE Ø8
[4V5]   113: Ø2 72 F3 EB D2 A2 BC Ø6 B1 2Ø BA 8C ØA B4 3C CD
[WM5]   114: 21 A3 F6 Ø6 73 Ø2 EB BF 33 FF B9 ØØ Ø2 BØ ØØ E8
[EY5]   115: 82 FF E2 F9 AC AD A3 C1 Ø6 AC F6 E7 B3 ØØ 8A F8
[946]   116: AD 8B C8 AC A2 DF Ø6 AD FF DØ BE D8 Ø6 E8 D3 FE
[UP5]   117: E8 68 FF B8 Ø2 42 8B 1E F6 Ø6 33 C9 33 D2 CD 21
[AE5]   118: 72 C4 5Ø 25 FF Ø1 75 Ø3 B8 ØØ Ø2 A3 3B Ø6 58 5Ø
[FK5]   119: 52 Ø5 FF Ø1 83 D2 ØØ B9 ØØ Ø2 F7 F1 A3 3D Ø6 5A
[KL5]   12Ø: 58 2D F1 Ø1 83 DA ØØ B9 1Ø ØØ F7 F1 A3 47 Ø6 B8
[TM5]   121: ØØ 42 33 C9 33 D2 CD 21 72 8C BE 37 Ø6 E8 83 FE
[RT5]   122: E8 18 FF 8B 1E F6 Ø6 B4 3E CD 21 73 Ø3 E9 37 FF
[TØ6]   123: BE 8C ØA AC 3C ØØ 75 FB FD AC 81 FE 8C ØA 72 1C
[1R5]   124: AC 3C 5C 74 17 3C 3Ø 72 F1 3C 39 77 ED FE CØ 88
[DH5]   125: 44 Ø1 3C 39 76 Ø6 C6 44 Ø1 3Ø EB DE C3 F4 Ø8 ØØ
[C45]   126: ØØ 5B Ø9 ØØ ØØ FB Ø9 ØØ ØØ F4 Ø8 ØØ ØØ ØØ ØA ØØ
[K65]   127: ØØ 14 ØA ØØ ØØ 3E ØA ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ FF
[3E5]   128: ØØ ØØ ØØ ØØ 5Ø E4 6Ø 3C 38 75 Ø8 2E C6 Ø6 F2 Ø8
[TR5]   129: FF EB 41 3C B8 75 Ø8 2E C6 Ø6 F2 Ø8 ØØ EB 35 3C
[WH5]   13Ø: 54 74 ØC 3C 37 75 2D 2E 8Ø 3E F2 Ø8 ØØ 74 25 2E
[FP5]   131: 8Ø 3E F3 Ø8 ØØ 75 ØD 2E C6 Ø6 F3 Ø8 FF 56 57 E8
[PK5]   132: 29 Ø1 5F 5E E4 61 ØC 8Ø E6 61 24 7F E6 61 BØ 61
[VL5]   133: E6 2Ø 58 CF 58 2E FF 2E CD Ø8 ØØ Ø1 Ø2 Ø3 Ø4 Ø5
[AG5]   134: 14 Ø7 38 39 3A 3B 3C 3D 3E 3F ØØ 22 E4 74 ØF 8Ø
[RN5]   135: FC ØB 74 33 8Ø FC 1Ø 73 41 2E FF 2E D1 Ø8 2E C6
[CA5]   136: Ø6 1A Ø2 Ø1 2E C6 Ø6 34 Ø2 1Ø 2E C6 Ø6 3B Ø2 ØØ
[XL5]   137: 5Ø 53 BB 1Ø ØØ 2E 8A 87 4A Ø9 2E 88 87 FB Ø3 FE
[3S5]   138: CB 79 F2 5B 58 EB D2 8Ø FF ØØ 75 Ø7 2E 88 1E 34
[NH5]   139: Ø2 EB C6 2E 88 1E 3B Ø2 EB BF 3D ØØ 1Ø 75 14 8Ø
[GT5]   14Ø: FB 1Ø 73 B5 57 8B FB 81 E7 FF ØØ 2E 88 BD FB Ø3
[ØR5]   141: 5F EB A6 3D Ø1 1Ø 75 Ø7 2E 88 3E ØB Ø4 EB 9A 3D
[LG5]   142: Ø2 1Ø 75 1A 5Ø 53 57 8B FA BB 1Ø ØØ 26 8A Ø1 2E
[LU5]   143: 88 87 FB Ø3 FE CB 79 F4 5F 5B 58 E9 7B FF 3D Ø3
[WM5]   144: 1Ø 75 Ø5 2E 88 1E 1A Ø2 E9 6E FF E8 43 ØØ CD 19
[4R5]   145: 9C 2E FF 1E DD Ø8 1E 56 2E C5 36 E9 Ø8 8Ø 3C ØØ
[PE5]   146: 5E 1F 75 29 5Ø 53 51 52 56 57 55 1E Ø6 E8 3B ØØ
[3S5]   147: FB E8 4E ØØ 8C C8 8E CØ E8 B4 FD E8 44 ØØ Ø7 1F
[9R5]   148: 5D 5F 5E 5A 59 5B 58 2E C6 Ø6 F3 Ø8 ØØ CF BØ Ø3
[HR5]   149: CF BF 24 ØØ BE CD Ø8 E8 2E ØØ BF 4Ø ØØ BE D1 Ø8
[TM5]   15Ø: E8 25 ØØ BF 64 ØØ BE D5 Ø8 EB 1D 1E Ø6 BF 2Ø ØØ
[ØN5]   151: BE DD Ø8 E8 12 ØØ BF AØ ØØ BE E1 Ø8 E8 Ø9 ØØ Ø7
[ZX5]   152: 1F C3 BF 9Ø ØØ BE E5 Ø8 FC 8C C8 8E D8 33 CØ 8E
[5G5]   153: CØ E8 ØØ ØØ 26 8B Ø5 A5 89 44 FE C3 42 49 4C 44
[ØR5]   154: 3Ø 2E 45 58 45 ØØ BE 6C ØC E8 A1 Ø1 BF CF Ø8 8C
[9P5]   155: C8 B9 Ø7 ØØ AB 83 C7 Ø2 E2 FA B8 Ø9 35 CD 21 BE
[8U5]   156: F4 Ø8 8B FB B9 2Ø ØØ F3 A7 74 18 BE 82 ØC E8 7C
[9V5]   157: Ø1 FA E8 7C FF B4 34 CD 21 89 1E E9 Ø8 8C Ø6 EB
[MM5]   158: Ø8 ØE Ø7 BE 8Ø ØØ AC B4 ØØ Ø3 FØ C6 Ø4 ØØ BE 81
[YG5]   159: ØØ AC 3C ØØ 74 17 3C 2Ø 74 F7 3C 2F 74 3Ø BF 8C
[FS5]   16Ø: ØA B4 4Ø FE CC 75 Ø9 BE B5 ØC E8 4Ø Ø1 E9 B6 ØØ
[6M5]   161: 3C ØØ 74 F9 3C 2F 74 16 3C 2Ø 74 1F 3C 61 72 Ø6
[SS5]   162: 3C 7A 77 Ø2 24 DF AA 26 C6 Ø5 ØØ AC EB D5 AC ØC
[DP5]   163: 2Ø 3C 61 75 Ø9 26 C6 Ø6 EF Ø8 FF EB 76 9Ø 3C 68
[KL5]   164: 75 21 26 C6 Ø6 FØ Ø8 FF AC 3C 2C 75 67 AC 3C 3Ø
[8G5]   165: 74 Ø7 3C 31 74 Ø3 E9 BØ ØØ 2C 3Ø 26 A2 F1 Ø8 EB
[3P5]   166: 52 EB CB 3C 6D 75 Ø8 26 C6 Ø6 FØ Ø8 ØØ EB 44 E8
[3G5]   167: E1 ØØ 72 39 B1 Ø4 D2 EØ 5Ø AC ØC 2Ø E8 D4 ØØ 59
[TF5]   168: 72 2B ØA C1 BB Ø3 Ø1 26 3A Ø7 74 ØE 83 C3 Ø9 81
[HP5]   169: FB Ø8 Ø2 72 F2 BE 34 ØD EB 29 26 A2 ED Ø8 26 C6
[5L5]   17Ø: Ø6 EF Ø8 ØØ AC 3C 2C 75 ØB AC E8 A6 ØØ 72 5A 26
[4S5]   171: A2 EE Ø8 AC 3C ØØ 74 ØE 3C 2Ø 74 F7 3C 2F 74 A1
[7V5]   172: BE 6A ØD E8 87 ØØ FB BE 9B ØD E8 8Ø ØØ 1E Ø6 1F
[ZR5]   173: BE 8C ØA E8 77 ØØ 1F BE B5 ØD E8 7Ø ØØ 26 8Ø 3E
[CN5]   174: EF Ø8 ØØ 75 29 26 8A 16 ED Ø8 52 B1 Ø4 D2 EA E8
[9M5]   175: 77 ØØ 5A 8Ø E2 ØF E8 7Ø ØØ BE CF ØD E8 4E ØØ 26
[7Z5]   176: 8A 16 EE Ø8 E8 62 ØØ EB ØB BE ED ØC EB B5 BE D8
[DF5]   177: ØD E8 39 ØØ BE 2C ØE E8 33 ØØ 26 8Ø 3E FØ Ø8 ØØ
[7K5]   178: 74 14 BE EF ØD E8 25 ØØ 26 8A 16 F1 Ø8 E8 39 ØØ
[6T5]   179: BE 2Ø ØE E8 17 ØØ 8C C8 8C C3 3B C3 75 Ø5 BA CC
[PP5]   18Ø: ØA CD 27 B4 4C CD 21 8A DØ B4 Ø6 CD 21 AC 22 CØ
[ZH5]   181: 75 F5 C3 2C 3Ø 72 1Ø 3C ØA 72 ØA 3C 3Ø 76 Ø8 2C
[UL5]   182: 27 3C ØF 77 Ø2 F8 C3 F9 C3 8Ø E2 ØF 8Ø C2 3Ø 8Ø
[6G5]   183: FA 39 76 Ø3 8Ø C2 Ø7 B4 Ø2 CD 21 C3 ØA 53 6E 61
[ØK5]   184: 7Ø 53 68 6F 74 2Ø 56 65 72 73 69 6F 6E 2Ø 32 2E
[5E5]   185: 37 ØØ 2Ø 69 6E 73 74 61 6C 6C 69 65 72 74 2E 2Ø
[GG5]   186: 41 75 66 72 75 66 2Ø 6D 69 74 2Ø 3C 41 6C 74 2D
[XC5]   187: 5Ø 72 74 53 63 3E 2Ø 62 7A 77 2E 2Ø 3C 53 79 73
[PD5]   188: 52 65 71 3E ØØ ØD ØA 46 65 68 6C 65 72 3A 2Ø 44
[HE5]   189: 65 72 2Ø 61 6E 67 65 67 65 62 65 6E 65 2Ø 5Ø 66
[5F5]   19Ø: 61 64 2Ø 69 73 74 2Ø 6C 84 6E 67 65 72 2Ø 61 6C
[5D5]   191: 73 2Ø 36 32 2Ø 5A 65 69 63 68 65 6E ØØ ØD ØA 46
[9C5]   192: 65 68 6C 65 72 3A 2Ø 45 72 6C 61 75 62 74 65 2Ø
[MN5]   193: 4F 7Ø 74 69 6F 6E 65 6E 2Ø 73 69 6E 64 2Ø 2F 4D
[2K5]   194: 6F 64 75 73 5B 2C 53 65 69 74 65 5D 2C 2Ø 2F 41
[4H5]   195: 2C 2Ø 2F 48 5B 2C 53 65 69 74 65 5D 2Ø 75 6E 64
[FE5]   196: 2Ø 2F 4D ØØ ØD ØA 46 65 68 6C 65 72 3A 2Ø 44 65
[SD5]   197: 72 2Ø 61 6E 67 65 67 65 62 65 6E 65 2Ø 56 69 64
[VG5]   198: 65 6F 6D 6F 64 75 73 2Ø 69 73 74 2Ø 6E 69 63 68
[LH5]   199: 74 2Ø 62 65 6B 61 6E 6E 74 ØØ ØD ØA 46 65 68 6C
[MA5]   2ØØ: 65 72 3A 2Ø 53 79 6E 74 61 78 2Ø 69 73 74 2Ø 53
[9G5]   2Ø1: 6E 61 7Ø 73 68 6F 74 2Ø 5B 5Ø 66 61 64 5D 2Ø 5B
[SK5]   2Ø2: 2F 4F 7Ø 74 69 6F 6E 65 6E 5D ØØ ØD ØA 2Ø 2Ø 41
[7F5]   2Ø3: 6B 74 75 65 6C 6C 65 72 2Ø 44 61 74 65 69 6E 61
[8D5]   2Ø4: 6D 65 3A 2Ø ØØ ØD ØA 2Ø 2Ø 45 69 6E 67 65 73 74
[CH5]   2Ø5: 65 6C 6C 74 65 72 2Ø 4D 6F 64 75 73 3A 2Ø ØØ 2C
[ØE5]   2Ø6: 2Ø 53 65 69 74 65 2Ø ØØ 41 75 74 6F 6D 61 74 69
[TE5]   2Ø7: 73 63 68 65 2Ø 45 72 6B 65 6E 6E 75 6E 67 ØØ 2Ø
[8C5]   2Ø8: 2Ø 49 6D 2Ø 4D 6F 64 75 73 2Ø 3Ø 37 2Ø 77 69 72
[AD5]   2Ø9: 64 2Ø 64 69 65 2Ø 48 65 72 63 75 6C 65 73 67 72
[7B5]   21Ø: 61 66 69 6B 2Ø 61 75 66 2Ø 53 65 69 74 65 2Ø ØØ
[SY4]   211: 2Ø 67 65 73 7Ø 65 69 63 68 65 72 74 ØD ØA ØØ
```

Listing 2. Wenn Sie keinen Assembler besitzen, dann tippen Sie das Hexdump »snapshot.com« ab

Name	palette.pas
Sprache	Turbo-Pascal ab Version 4.Ø
Grafikkarte	jede
Besonderheiten	Verändert Snapshot-Bilder

```
[AS4]     1: {$R-,S-,I-,D-,F-,V-,B-,M 8192,Ø,Ø}
[ASØ]     2: Uses
[9G1]     3:   Crt, Dos;
[ØØØ]     4:
[RNØ]     5: Var
[XD3]     6:   GrafikFlag: Boolean;
[Z53]     7:   Datei: File Of Byte;
[LW2]     8:   Bildname: String;
[NL7]     9:   VideoModus, Wert, Hintergrund, Palette: B
              yte;
[3S2]    1Ø:   Reg: Registers;
[WY1]    11:   Taste: Char;
[SH6]    12:   Rot, Gruen, Blau: Array [Ø..255] Of Byte;
[UØ4]    13:   Farben, Max, I: Integer;
[983]    14:   Blinkflag: Boolean;
[ØØØ]    15:
[XV5]    16: Procedure Fehler (Meldung: String);
[MUØ]    17:   Begin
[KF6]    18:     If GrafikFlag Then TextMode (LastMode);
[AS4]    19:     Writeln ('Fehler: ', Meldung);
[W31]    2Ø:     Halt (1)
[ZNØ]    21:   End;
[ØØØ]    22:
[PP7]    23: Procedure SetzePosition (Position: LongInt)
              ;
[MUØ]    24:   Begin
[V84]    25:     Seek (Datei, Position - 64);
[E7A]    26:     If IOResult <> Ø Then Fehler ('Datei ka
              nn nicht gelesen werden')
[ZNØ]    27:   End;
[ØØØ]    28:
[6H3]    29: Procedure ZeigeBild;
[MUØ]    3Ø:   Begin
[U23]    31:     Exec (Bildname, 'C');
[BHA]    32:     If DosExitCode <> 6 Then Fehler ('Bild
              kann nicht geladen werden');
[R43]    33:     GrafikFlag := True;
[753]    34:     SetzePosition (692)
[ZNØ]    35:   End;
[ØØØ]    36:
[1C2]    37: Procedure CGA;
[MUØ]    38:   Begin
[7R1]    39:     ZeigeBild;
[X94]    4Ø:     Read (Datei, Hintergrund);
```

```
[R73]   41:      SetzePosition (699);
[9H3]   42:      Read (Datei, Palette);
[PH8]   43:      Palette := (Palette And 1) Or (Hintergr
         und Shr 3 And 2);
[P31]   44:      Repeat
[XT9]   45:        Hintergrund := (Hintergrund And 15) O
         r (Palette Shl 3 And 16);
[ØX1]   46:        Reg.AH := $ØB;
[3N3]   47:        Reg.BX := Hintergrund;
[B42]   48:        Intr ($1Ø, Reg);
[ØX1]   49:        Reg.AH := $ØB;
[ZR4]   5Ø:        Reg.BX := 256 Or (Palette And 1);
[B42]   51:        Intr ($1Ø, Reg);
[HR2]   52:        Taste := ReadKey;
[Z12]   53:        Case Taste Of
[6P3]   54:          'h': Inc (Hintergrund);
[LM3]   55:          'H': Dec (Hintergrund);
[ZY2]   56:          'p': Inc (Palette);
[CX2]   57:          'P': Dec (Palette);
[ZEØ]   58:          #13:
[MUØ]   59:            Begin
[273]   6Ø:              SetzePosition (692);
[TF4]   61:              Write (Datei, Hintergrund);
[R73]   62:              SetzePosition (699);
[ØU3]   63:              Palette := Palette And 1;
[EG3]   64:              Write (Datei, Palette)
[4MØ]   65:            End
[4MØ]   66:      End
[PS3]   67:      Until Taste In [#13, #27]
[ZNØ]   68:    End;
[ØØØ]   69:
[LK3]   7Ø: Procedure HoleTaste;
[MUØ]   71:    Begin
[HR2]   72:      Taste := ReadKey;
[LS4]   73:      Blinkflag := Taste In [#Ø, ' '];
[EN2]   74:      If Taste = #Ø Then
[ØM2]   75:        Case ReadKey Of
[YP1]   76:          #$4D: Inc (I);
[ZN1]   77:          #$4B: Dec (I);
[XX1]   78:          #$74: I := I + 8;
[2Y1]   79:          #$73: I := I - 8;
[2H1]   8Ø:          #$47: I := Ø;
[2V1]   81:          #$4F: I := Max
[ZNØ]   82:        End;
[TG3]   83:      If I > Max Then I := Max;
[3S2]   84:      If I < Ø Then I := Ø;
[6T2]   85:      If Taste = 'Ø' Then
[MUØ]   86:        Begin
[R32]   87:          Rot [I] := Ø;
[9E2]   88:          Gruen [I] := Ø;
[NZ1]   89:          Blau [I] := Ø
[ZNØ]   9Ø:        End;
[5EB]   91:      If (Taste In ['r', 'm', 'y', 'w']) And
         (Rot [I] < Farben) Then Inc (Rot [I]);
[V8A]   92:      If (Taste In ['R', 'M', 'Y', 'W']) And
         (Rot [I] > Ø) Then Dec (Rot [I]);
[G2C]   93:      If (Taste In ['g', 'y', 'c', 'w']) And
         (Gruen [I] < Farben) Then Inc (Gruen [I]);
[PXA]   94:      If (Taste In ['G', 'Y', 'C', 'W']) And
         (Gruen [I] > Ø) Then Dec (Gruen [I]);
[4FB]   95:      If (Taste In ['b', 'c', 'm', 'w']) And
         (Blau [I] < Farben) Then Inc (Blau [I]);
[Ø6A]   96:      If (Taste In ['B', 'C', 'M', 'W']) And
         (Blau [I] > Ø) Then Dec (Blau [I])
[ZNØ]   97:    End;
[ØØØ]   98:
[9C2]   99: Procedure EGA;
[GL3]   1ØØ:   Function Farbe: Byte;
[MUØ]   1Ø1:     Begin
[FG8]   1Ø2:       Farbe := (Rot [I] Shl 5 And 32) Or (R
         ot [I] Shl 1 And 4)
[LA7]   1Ø3:             Or (Gruen [I] Shl 4 And 16) Or (Gru
         en [I] And 2)
[ZB7]   1Ø4:             Or (Blau [I] Shl 3 And 8) Or (Blau
         [I] Shr 1 And 1)
[ZNØ]   1Ø5:       End;
[MUØ]   1Ø6:     Begin
[7R1]   1Ø7:       ZeigeBild;
[WP2]   1Ø8:       For I := Ø To 16 Do
[MUØ]   1Ø9:         Begin
[523]   11Ø:           Read (Datei, Wert);
[DR7]   111:           Rot [I] := (Wert Shr 5 And 1) Or (W
         ert Shr 1 And 2);
[K57]   112:           Gruen [I] := (Wert Shr 4 And 1) Or
         (Wert And 2);
[AP7]   113:           Blau [I] := (Wert Shr 3 And 1) Or (
         Wert Shl 1 And 2)
[ZNØ]   114:         End;
[8R1]   115:       Farben := 3;
[PE1]   116:       Max := 16;
[LYØ]   117:       I := Ø;
[P31]   118:       Repeat
[EX6]   119:         If I < 16 Then Reg.AX := $1ØØØ Else R
         eg.AX := $1ØØ1;
[SS1]   12Ø:         Reg.BL := I;
[ZD2]   121:         Reg.BH := Farbe;
[B42]   122:         Intr ($1Ø, Reg);
[ST1]   123:         HoleTaste;
[416]   124:         If Blinkflag Then For Wert := Ø To 3
         Do
[MUØ]   125:           Begin
[EX6]   126:             If I < 16 Then Reg.AX := $1ØØØ El
         se Reg.AX := $1ØØ1;
[SS1]   127:             Reg.BL := I;
[ØX3]   128:             Reg.BH := (Wert And 1) * 63;
[G22]   129:             Intr ($1Ø, Reg)
[4MØ]   13Ø:           End
[7Y3]   131:       Until Taste In [#13, #27];
[8U2]   132:       If Taste = #13 Then
[MUØ]   133:         Begin
[273]   134:           SetzePosition (692);
[WP2]   135:           For I := Ø To 16 Do
[MUØ]   136:             Begin
[F92]   137:               Wert := Farbe;
[233]   138:               Write (Datei, Wert)
[4MØ]   139:             End
[4MØ]   14Ø:         End
[ZNØ]   141:    End;
[ØØØ]   142:
[DE2]   143: Procedure VGA;
[MUØ]   144:    Begin
[7R1]   145:      ZeigeBild;
[AE9]   146:      For I := Ø To 255 Do Read (Datei, Rot [
         I], Gruen [I], Blau [I]);
[2V1]   147:      Farben := 63;
[FN1]   148:      Max := 255;
[LYØ]   149:      I := Ø;
[P31]   15Ø:      Repeat
[A42]   151:        Reg.AX := $1Ø1Ø;
[SS1]   152:        Reg.BL := I;
[GV2]   153:        Reg.DH := Rot [I];
[E93]   154:        Reg.CH := Gruen [I];
[ØØ3]   155:        Reg.CL := Blau [I];
[B42]   156:        Intr ($1Ø, Reg);
[ST1]   157:        HoleTaste;
[XZ5]   158:        If Blinkflag Then For Wert := Ø To 31
         Do
[MUØ]   159:          Begin
[A42]   16Ø:            Reg.AX := $1Ø1Ø;
[WT1]   161:            Reg.BX := I;
[4X3]   162:            Reg.DH := (Wert And 1) * 63;
[2X3]   163:            Reg.CH := (Wert And 1) * 63;
[EX3]   164:            Reg.CL := (Wert And 1) * 63;
[B42]   165:            Intr ($1Ø, Reg);
[X71]   166:            Delay (1)
[4MØ]   167:          End
[7Y3]   168:      Until Taste In [#13, #27];
[8U2]   169:      If Taste = #13 Then
[MUØ]   17Ø:        Begin
[273]   171:          SetzePosition (692);
[PT9]   172:          For I := Ø To 255 Do Write (Datei,
         Rot [I], Gruen [I], Blau [I])
[4MØ]   173:        End
[ZNØ]   174:    End;
[ØØØ]   175:
[MUØ]   176: Begin
[1F2]   177:    { Version 1.Ø1 }
[963]   178:    GrafikFlag := False;
[DVA]   179:    If ParamCount <> 1 Then Fehler ('Syntax i
         st "Palette <Bildname>"');
[XT4]   18Ø:    If Pos ('.', ParamStr (1)) > Ø Then
[LV3]   181:      Bildname := ParamStr (1)
[ZRØ]   182:    Else
[A55]   183:      Bildname := ParamStr (1) + '.EXE';
[SY3]   184:    Assign (Datei, Bildname);
[E62]   185:    Reset (Datei);
[AF8]   186:    If IOResult <> Ø Then Fehler ('Datei nich
         t gefunden');
[L73]   187:    SetzePosition (679);
[154]   188:    Read (Datei, VideoModus);
[173]   189:    Case VideoModus Of
[FP1]   19Ø:      $Ø4, $Ø5: CGA;
[7B3]   191:      $ØD, $ØE, $1Ø, $12, $37: EGA;
[GA2]   192:      $13, $2D, $2E: VGA;
[CA6]   193:      Else Fehler ('Unbekanntes Bildformat')
[ZNØ]   194:    End;
[552]   195:    Close (Datei);
[BC3]   196:    TextMode (LastMode);
[VH8]   197:    If Taste = #27 Then Writeln ('Abbruch dur
         ch Benutzer')
[KNØ]   198: End.
```

Listing 3. Mit »palette.pas« ändern Sie nachträglich die Farben jedes Snapshot-Bildes

Kopieren mit 115200 Bit pro Sekunde

Sichere Datenübertragung über die serielle Schnittstelle

Wenn Sie Daten von einem Rechner auf einen anderen übertragen möchten, bietet Ihnen MS-DOS leider nur unzureichende Möglichkeiten, weil die im BIOS implementierte Treiber-Software für die RS232-Schnittstelle einige Mängel aufweist. Ohne zusätzliches Software-Protokoll oder Einsatz eines Terminalprogramms ist an eine fehlerfreie Datenübertragung kaum zu denken. Ein RTS/CTS-Hardware-Protokoll schafft hier Abhilfe und bietet außerdem die Möglichkeit, mit angemessen hohen Baud-Raten arbeiten zu können.

Die serielle Datenübertragung zählt sicherlich zu den am häufigsten angewandten Methoden, Software zwischen Rechnern mit verschiedenen Systemen hin und her zu kopieren. Selbst für die Verbindung zweier PCs spielt sie eine wichtige Rolle, wenn der eine Rechner beispielsweise über ein 5 1/4-Zoll-, der andere über ein 3 1/2-Zoll-Diskettenlaufwerk verfügt.

Als Minimalkonfiguration für die Übertragung genügt ein 5poliges Kabel für die Leitungen RD und TD (Datenübermittlung), RTS und CTS (Handshake-Protokoll) sowie GND (Masse). Was Sie nun noch benötigen, sind eine neue RS232-Treiber-Software und der MS-DOS-Befehl »copy«. Durch das hier abgedruckte Utility »com2bin« ist auch der Empfang binärer Daten, zum Beispiel ausführbarer Programme, möglich.

Was geschieht nun beim Empfangen der Dateien? Über das LSR (Line-Status-Register des 8250) erkennt das Com2bin-Programm, ob gerade Daten an der Schnittstelle anliegen. In diesem Fall liest es diese aus dem RBR (Receive-Buffer-Register), nachdem die RTS-Leitung inaktiviert wurde.

Wenn der PC hingegen bei inaktivierter RTS-Leitung ein Zeichen empfängt, kann dies zweierlei bedeuten: Zum einen kann es sein, daß sich der Sender nicht an das Hardware-Protokoll hält. Dies ist schlecht, weil Sie dann ohne Terminalprogramm nicht fehlerfrei übertragen können. Zum anderen besteht die Möglichkeit, daß der Sender verspätet auf den RTS-Pegel reagiert, was sich aber durch ein Hardware-Handshake vermeiden läßt.

Normalerweise aktiviert das Programm die RTS-Leitung über das MCR (Modem Control Register), um dem Sender Empfangsbereitschaft anzuzeigen. Nun wartet der Rechner so lange, bis entweder ein Zeichen empfangen wird, eine eventuelle Zeitbegrenzung abgelaufen ist oder der Anwender [Ctrl-Break] gedrückt hat. Nach dem Deaktivieren der RTS-Steuerleitung wird das Zeichen zur Weiterverarbeitung ausgelesen.

Wesentlich unkomplizierter als das Empfangen ist das Senden von Daten. Hier prüft die Routine über das MSR (Modem-Status-Register) den Status der CTS-Leitung so lange, bis der Empfänger seine Bereitschaft zur Datenaufnahme zeigt. Diese Warteschleife kann zeitlich begrenzt sein und von Ihnen jederzeit durch [Ctrl-Break] unterbrochen werden. Zur Sicherheit prüft der Rechner noch, ob der Ausgabepuffer des Schnittstellenbausteins 8250 leer ist und durchläuft gegebenenfalls eine weitere Warteschleife. Dieses Verfahren stellt sicher, daß auch dann keine Zeichen bei der Übertragung verlorengehen, wenn der Empfänger verspätet die CTS-Leitung deaktiviert. Ist der Empfänger zur Aufnahme bereit, dann werden ihm die Daten über das THR (Transmitter-Hold-Register) gesendet.

Die Konfiguration der seriellen Schnittstelle sieht recht einfach aus, weil Sie beim Programmieren ohnehin gewisse Regeln einhalten müssen. Beispielsweise wird anstelle der Baud-Rate im Klartext dem 8250 das Ergebnis der Operation

```
int (115200 div <Baud-Rate>)
```

übergeben, woran Sie zwei wichtige Dinge erkennen: Es sind keine beliebigen Baud-Raten möglich, und die maximal zulässige Baud-Rate beträgt 115200 Baud. Die nächstkleinere ist 57600.

Diese theoretischen Überlegungen sind in dem neuen Schnittstellentreiber »nmode.asm« (Listing 1) und dem Übertragungsprogramm »com2bin.c« (Listing 2) realisiert. Ein kleiner Schönheitsfehler des neuen Treibers sollte hier aber nicht unerwähnt bleiben. Der MS-DOS-Befehl »mode« mit Parameter »p« (Zeitbegrenzung ausschalten) deaktiviert den Nmode-Treiber, weil dadurch die Interrupt-Vektoren 14hex und 17hex ohne Rücksicht auf andere Programme (hier: Nmode) auf ihren Standardwert zurückgesetzt werden.

Software, die nicht den Interrupt 14hex zur Abfrage der seriellen Schnittstelle verwendet, wird natürlich nicht vom neuen Treiber profitieren können. Hierzu zählen die meisten Terminalprogramme, die eigene Methoden zur Ansteuerung der seriellen Schnittstelle einsetzen und damit leider auch keinen Vorteil aus dem Hardware-Protokoll ziehen. Allerdings besteht der Sinn von Nmode und Com2bin ja auch darin, den Datentransfer auf der MS-DOS-Ebene zu praktizieren.

Der Aufruf von Nmode gestaltet sich für Sie sehr einfach, weil die Parameter denen des MS-DOS-Befehls »mode« gleichen. Im folgenden Text und auch bei der Beschreibung von Com2bin bedeuten Parameter in eckigen Klammern, daß die Angabe nicht zwingend notwendig ist und gegebenenfalls durch einen vorgegebenen Wert ersetzt wird (»default«). Die Klammern dürfen Sie selbstverständlich nicht eingeben.

So konfigurieren Sie die serielle Schnittstelle:

```
nmode {COMx | SERx | AUX} [:] <Baud> [,<Parität> [,<Datenbits> [,<Stopbits> [,P] ] ] ]
```

Anstelle von »x« setzen Sie einen Wert zwischen 1 und 4 ein. Je nach Wunsch können Sie die Schlüsselworte »com« oder »ser« angeben. »Com1« ist dabei gleichbedeutend mit »aux«.

Die Baud-Rate dürfen Sie wie folgt abkürzen:

```
11 = 110 Baud      24 = 2400 Baud
15 = 150 Baud      48 = 4800 Baud
30 = 300 Baud      96 = 9600 Baud
60 = 600 Baud      19 = 19200 Baud
12 = 1200 Baud     38 = 38400 Baud
```

Alle anderen Baud-Raten müssen Sie in voller Länge angeben. Allerdings ist aus technischen Gründen nicht jede beliebige Baud-Rate möglich. Deshalb gibt das Nmode-Programm nach der Konfiguration die tatsächlich eingestellte Baud-Rate aus. Falls Sie einen ungültigen Wert gewählt haben, rundet die Routing Ihre Angabe zur nächsthöheren Geschwindigkeit auf. Die vorgegebenen Konfigurationsparameter sind »e,7,1« (gerade Parität, 7 Daten- und 1 Stopbit).

Wenn Sie zusätzlich den Parameter »p« setzen, so bricht das Programm bei Fehlern nicht nach einer bestimmten Zeit ab. Die Routine ist erst beendet, wenn das Zeichen erfolgreich gesendet werden konnte oder Sie [Ctrl-Break] gedrückt haben.
Die MS-DOS-Dienstprogramme Mode und Nmode dürfen von Ihnen nicht abwechselnd aufgerufen werden. In Nmode sind daher die Konfigurationsparameter für eine Umleitung der parallelen auf die serielle Schnittstelle bereits integriert. Die ab MS-DOS 3.3 gegebenen Möglichkeiten einer Zeichensatztabellenzuordnung (»codepages«) sind in Nmode jedoch nicht implementiert. Es empfiehlt sich daher, bei Nutzung von Nmode auf die Codepages zu verzichten.
Wenn Sie selbst im Quelltext Änderungen vornehmen möchten, dann werden Ihnen die folgenden Erklärungen nützlich sein:
Das Programm ist für den A86-Assembler geschrieben. Allerdings wurde Wert auf leichte Übertragbarkeit für den MASM-Assembler gelegt.
Bei Nmode und Com2bin wurden alle Möglichkeiten des Prozessors ausgenutzt sowie intensiv mit den Prozessor-Flags gearbeitet (speziell zur Anzeige von Fehlern). Wenn Sie den Code also ändern wollen, sollten Sie sich zuvor einen genauen Überblick über die Interpretation der Flags und Registerinhalte verschaffen.
Im Parser-Teil von Nmode zeigt das DI-Register auf die MS-DOS-Eingabezeile und SI auf den Beginn des Parameterblocks. Änderungen dieser Register sollten sorgfältig erfolgen.
Im Konfigurationsteil stehen relativ häufig sogenannte »segment overrides«. Im allgemeinen verweist CS auf das Programm Nmode und seine Daten, DS auf das BIOS-Segment 0 und ES auf den Schnittstellentreiber selbst.
Das Programm Nmode installiert die Schnittstellen-Software resident. Dabei verbleibt der Code zwischen »org 100h« und dem Label »endresident« im Speicher.
Wie wird der neue Schnittstellentreiber nun eingesetzt? Da der Copy-Befehl keine Binärdatei von der seriellen Schnittstelle einlesen kann (»copy com1 datei.ext«), wurde das Utility Com2bin (Listing 2) geschrieben. Entwickelt wurde dieses Programm mit dem PCC-Compiler (Desmet-C). Weil der Autor allerdings eine erweiterte Library verwendet, die sich eng an den Turbo-C-Dialekt anlehnt, sollte es keine Probleme bei der Anpassung geben. Wer allerdings über einen Desmet-C-Compiler verfügt, muß die Peek-, Poke- und Int86-Aufrufe umschreiben.
Der Aufruf zum Übersetzen und Linken gestaltet sich wie folgt:

```
com2bin <Dateiname> [P<Portnummer>] [T<Zeitbegrenzung>]
```

Damit bereiten Sie einen der beiden Rechner zum Empfang einer Datei vor. Über den Parameter »p«, gefolgt von einer Zahl von 1 bis 4, wählen Sie die serielle Schnittstelle, von der gelesen werden soll. Die Voreinstellung ist 1 für »com1«.
Der Parameter »t« bezeichnet die Anzahl der Wiederholungen bei Auftreten eines Fehlers. Bei »t0« wartet das Programm so lange, bis Sie es mit [Ctrl-Break] abbrechen. Andererseits ist die Zeitdauer abhängig vom verwendeten Rechner, auf dem Com2bin läuft. Der Vorgabe »t1« entspricht etwa 1 Sekunde Wartezeit auf einem PC mit 4,77 MHz oder 0,5 Sekunden auf einem AT mit 8 MHz. Der Maximalwert ist »t255«.
Leider gibt es an einer seriellen Schnittstelle für ein Programm keine Möglichkeit, das Ende einer Binärdatei zu erkennen (es sei denn, Sie finden selbst eine Definition). Nach der Übertragung müssen Sie daher Com2bin mit [Ctrl-Break] beenden oder so lange warten, bis ein Fehler durch Zeitüberschreitung auftritt.
Datenübertragungsfehler werden zur Kontrolle auf dem Bildschirm angezeigt, das fehlerhafte Zeichen wird aber natürlich nicht in die Datei geschrieben. Sollten solche Fehler häufiger auftreten, dann überprüfen Sie das Datenkabel und das Protokoll auf den beteiligten Rechnern.
Das Com2bin-Programm macht bei einem Datenübertragungsfehler keine Unterscheidung zwischen den möglichen Fehlerarten (siehe Textkasten), lediglich ein Zeitfehler wird getrennt behandelt. Bei Bedarf können Sie das

Beschreibung der Register des Schnittstellenbausteins 8250

Zur Vervollständigung des Beitrags folgt an dieser Stelle eine kurze Beschreibung der Register des 8250-Bausteins, soweit dies der Erklärung des Schnittstellentreibers dient. Der UART 8250 ist der von IBM für die serielle Datenübertragung vorgesehene Baustein und wird auch bei kompatiblen PCs eingesetzt.
Die Beschreibung der Register des 8250 ist nicht vollständig, es fehlt zum Beispiel eine genauere Beschreibung des interruptgesteuerten Betriebs, der jedoch für einen reinen Datenaustausch nicht benötigt wird.

THR Transmitter-Hold-Register: enthält die Daten, die über die serielle Schnittstelle gesendet werden sollen

RBR Receive-Buffer-Register: beinhaltet das zuletzt eingelesene Byte von der seriellen Schnittstelle

DLL niederwertiges Byte der Baud-Rate

DLM höherwertiges Byte der Baud-Rate

IER Interrupt-Enable-Register: wird hier nicht verwendet

IIR Interrupt-Identification-Register: wird hier nicht benutzt

LCR Line-Control-Register:
Bit 0/1: (WLS0/WLS1) 10 = 7 Bit Datenlänge, 11 = 8 Bit Datenlänge
Bit 2: (STB) 0 = 1 Stopbit, 1 = 2 Stopbits
Bit 3/4/5: (PEN/EPS/STP) 001 = ungerade (odd) Parität, 011 = gerade (even) Parität, 000 = keine (none) Parität
Bit 6: Set Break (hier 0)
Bit 7: (DLAB) bestimmt, ob auf die Register THR beziehungsweise RBR und IER (0) oder auf DLL und DLM zugegriffen wird (1)

MCR Modem-Control-Register:
Bit 0: (DTR) Signalleitung »Data Terminal Ready«
Bit 1: (RTS) Signalleitung »Ready To Send«
Bit 2: (OUT1) immer 0
Bit 3: (OUT2) entspricht Interrupt-Enable (muß 0 sein)
Bit 4: (LOOP) immer 0
Die Bits 5 bis 7 müssen stets 0 sein.

LSR Line-Status-Register:
Bit 0: (DR) zeigt an, ob RBR ein Zeichen enthält
Bit 1: (ORE) Überlauffehler
Bit 2: (PE) Paritätsfehler
Bit 3: (FE) Fehler im Protokoll
Bit 4: (BI) Abbruch während des Empfangs eines Zeichens durch einen Interrupt
Bit 5: (THRE) gibt an, ob THR ein Zeichen enthält
Bit 6: (TEMT)
Bit 7: (UMG) Der Schnittstellentreiber setzt dieses Bit, um anzuzeigen, daß ein Fehler aufgetreten ist. Wenn kein Protokollfehler vorliegt, kann es sich nur um einen Ablauf der Warteschleife oder einen Abbruch durch [Ctrl-Break] handeln. Den zweiten Fall erkennen Sie durch Abfragen des Bit 7 der BIOS-Variablen 0040:0071hex.

MSR Modem-Status-Register:
Bit 0: (DCTS) zeigt an, ob sich der Zustand der CTS-Leitung seit dem letzten Lesen verändert hat
Bit 1: (DDSR) wie Bit 0, jedoch für die DSR-Leitung
Bit 2: (TERI) wie Bit 0, jedoch für die Ring-Indicator-Leitung
Bit 3: (DDCD) wie Bit 0, jedoch für die DCD-Leitung
Bit 4: (CTS) Signalleitung »Clear To Send«
Bit 5: (DSR) Signalleitung »Data-Set Ready«
Bit 6: (RI) Signalleitung »Ring Indicator«
Bit 7: (DCD) Signalleitung »Data Carrier Detect«

SCR Scratch-Register: wird hier nicht verwendet

Programm anhand der Fehlerliste jedoch in dieser Hinsicht komfortabler gestalten.

Das RTS/CTS-Hardware-Handshake ist die einfachste und bekannteste Methode, ein serielles Protokoll zu realisieren. Die RS232-Schnittstelle bietet aber noch eine Fülle weiterer Steuerleitungen. Dabei werden die Signale DTR, DSR und DCD am häufigsten verwendet. Das DCD-Signal spielt aber bei der Datenübertragung von Computer zu Computer keine Rolle und wird in dem hier gezeigten Programm auch nicht weiter berücksichtigt.

Wenn Sie zwei PCs miteinander verbinden, können Sie die Steuerleitungen selbst bestimmen und sollten daher auf das zuverlässige RTS/CTS-Protokoll zurückgreifen. Schwierig wird es hingegen, wenn der PC mit einem Rechner verbunden werden muß, der ein anderes Handshaking-Verfahren verwendet.

Damit Sie sich in diesem Fall nicht ein zusätzliches Übertragungskabel bauen müssen, wurde das Utility »confcom« (Listing 3) geschrieben, das eine beliebige Kombination aus RTS und DTR (für den Datenempfang) beziehungsweise CTS und DSR (Senden von Daten) wählen kann.

Nach dem Aufruf von Confcom geben Sie die serielle Schnittstelle (1 bis 4) an, deren Protokoll geändert werden soll. Sie können für jede verfügbare Schnittstelle ein eigenes Protokoll definieren. Anschließend bestimmen Sie die gewünschte Steuerleitung. Ein RTS/DTR-Protokoll bedeutet, daß beide Leitungen für den Empfang aktiviert werden. Das ist dann sinnvoll, wenn Sie das Protokoll des sendenden Rechners nicht kennen.

Schließlich geben Sie noch die Steuerleitung an, auf die beim Senden reagiert werden soll. Der Schnittstellentreiber spricht nur auf Signale an, die »getaktet« sind, das heißt, deren Zustand sich in einem fest vorgegebenen Zeitraum ändert. Dadurch ist es allerdings nicht möglich, Daten an ein Gerät zu senden, das CTS und DSR ständig auf Low- beziehungsweise High-Pegel hält. Sie beenden Confcom, indem Sie anstelle der Schnittstellennummer eine 0 eingeben.

Bei der Übertragung von PC zu PC muß natürlich auf beiden Rechnern der Nmode-Treiber installiert sein. Außerdem müssen beide Rechner mit denselben Parametern und dem gleichen Handshake-Protokoll arbeiten.

Problematisch ist die Dauer der Wiederholschleife. Die Vorgabe dieses Wertes von 1 Sekunde ist schon bei einem mit PC mit 4,77 MHz sehr knapp bemessen und sinkt bei schnelleren Rechnern leicht auf Werte unter 0,5 Sekunden.

Über das MS-DOS-Dienstprogramm »debug« können Sie den Wert bis um den Faktor 255 (FFhex) erhöhen:

```
debug
e 0040:007C FF FF FF FF
q
```

Die Adresse 7Chex steht für »com1«, 7Dhex für »com2«, 7Ehex für »com3« und 7Fhex für »com4«.

Bedenken Sie, daß Sie dem Zielrechner immer genügend Zeit einräumen sollten, die ankommenden Daten auch zu verarbeiten, also zu speichern. Deshalb ist es empfehlenswert, auf dem sendenden Rechner Nmode mit dem Parameter »p« zu aktivieren. In diesem Fall wiederholt der Treiber automatisch die Übertragung, sobald ein Zeitfehler auftritt.

Wenn allerdings der Zielrechner die Datenübertragung abbricht, müssen Sie die Endlosschleife auf dem sendenden Rechner durch [Ctrl-Break] beenden. Arbeiten Sie mit Zeitbegrenzung, können Sie auftretende Fehler wie beispielsweise

```
Gerätefehler COM1: ...,(W)iederholen,...?
```

einfach durch Drücken der W-Taste beheben.

Der Copy-Befehl von MS-DOS läßt dem sendenden Rechner jedoch genug Zeit, weil hier bei einem Fehler durch Zeitüberschreitung automatisch eine Wiederholung durchgeführt wird.

(Michael Stramm/wn)

Name	nmode.asm
Sprache	MASM, TASM
Grafikkarte	jede
Besonderheiten	keine

```
[7B6]    1: ;Autor: M.Stramm - geschr.: Aug/Sept '89
[5B6]    2: ;(c) RedaktionsbÜro Everts & Hagedorn
[ØØØ]    3:
[2T6]    4: ;assemblieren mit MASM oder kompatiblem
[6F7]    5: ;Assembler, anschließend  linken  und durch
[YL6]    6: ;EXE2BIN in ein .COM-File konvertieren.
[ØØØ]    7:
[5X1]    8:          name    nmode
[ØØØ]    9:
[AD2]   1Ø: code     segment
[ØØØ]   11:
[3X3]   12:          assume  cs:code,ds:code
[ØØØ]   13:
[AG1]   14:          org     1ØØh
[ØØØ]   15:
[RT2]   16: ComAdr   equ     4ØØh
[VU2]   17: ParAdr   equ     4Ø8h
[3U2]   18: BrkFlg   equ     471h
[X23]   19: ComWait  equ     47ch
[MH2]   2Ø: MxBaudH  equ     1
[B33]   21: MxBaudL  equ     49664
[ØØØ]   22:
[LM2]   23: RTS      equ     Ø2h
[GL2]   24: DTR      equ     Ø1h
[KL2]   25: CTS      equ     Ø1h
[GL2]   26: DSR      equ     Ø2h
[2D2]   27: DR       equ     Ø1h
```

```
[KM2]   28: THRE     equ     2Øh
[3K2]   29: DLAB     equ     8Øh
[ØØØ]   3Ø:
[FH4]   31: DHS      equ     RTS+(CTS SHL 2)
[ØØØ]   32:
[DS2]   33: SerWt    equ     8Øh
[LR2]   34: ParWt    equ     4Øh
[3V2]   35: ParSer   equ     2Øh
[V23]   36: SerPort  equ     Ø3h
[ØØØ]   37:
[H43]   38: start:   jmp     parse
[ØØØ]   39:
[R32]   4Ø: int17h   dd      Ø
[ØØØ]   41:
[3S2]   42: Flags    db      Ø,Ø,Ø,Ø
[VZ4]   43: HandShk  db      DHS,DHS,DHS,DHS
[ØØØ]   44:
[R54]   45: Baud     dw      417h     ;  11Ø Baud
[C92]   46:          dw      3ØØh     ;  15Ø
[S92]   47:          dw      18Øh     ;  3ØØ
[UA2]   48:          dw      Øc8h     ;  6ØØ
[5D2]   49:          dw      6Øh      ; 12ØØ
[5D2]   5Ø:          dw      3Øh      ; 24ØØ
[5E2]   51:          dw      18h      ; 48ØØ
[PF2]   52:          dw      Øch      ; 96ØØ
[YF2]   53:          dw      Ø6h      ;192ØØ
[CA3]   54:          dw      Ø3h      ;384ØØ Baud
[ØØØ]   55:
[2R2]   56: BaudCnt  equ     1Ø
[ØØØ]   57:
[ST2]   58: Ident    db      'NCOM'
[ØØØ]   59:
[591]   6Ø: int14h:
[6PØ]   61:      sti
[BK1]   62:      cmp     ah,4
[CT1]   63:      ja      end14h
[ØØØ]   64:
[ZF1]   65:      push    bx
[1G1]   66:      push    cx
```

```
[3G1]   67:       push    dx
[ØØØ]   68:
[MF1]   69:       push    ds
[BN1]   7Ø:       mov     bx,Ø
[2V1]   71:       mov     ds,bx
[LT1]   72:       mov     bl,dl
[B31]   73:       je      1Ø
[ØØØ]   74:
[7U3]   75:       mov     ch,[bx+ComWait]
[5R5]   76:       mov     cl,byte ptr cs:HandShk[bx]
[P36]   77:       test    byte ptr cs:Flags[bx],SerWt
[B31]   78:       je      1Ø
[ER1]   79:       sub     ch,ch
[GGØ]   8Ø: 1Ø:
[CR1]   81:       add     bx,bx
[XE3]   82:       mov     dx,[bx+ComAdr]
[GS1]   83:       and     dx,dx
[ED1]   84:       pop     ds
[B32]   85:       je      exit14h
[ØØØ]   86:
[LH1]   87:       add     dx,5
[9K1]   88:       cmp     ah,2
[531]   89:       jb      11
[531]   9Ø:       ja      12
[ØØØ]   91:
[WKØ]   92: ah2:
[6L1]   93:       in      al,dx
[NU1]   94:       test    al,DR
[2T1]   95:       mov     ah,al
[7F1]   96:       jne     131
[ØØØ]   97:
[LA1]   98:       dec     dx
[GT1]   99:       mov     al,cl
[5H1]   1ØØ:      and     al,3
[6V1]   1Ø1:      out     dx,al
[SB1]   1Ø2:      inc     dx
[1U1]   1Ø3:      mov     cl,DR
[P32]   1Ø4:      call    timing
[FHØ]   1Ø5: 131:
[LA1]   1Ø6:      dec     dx
[5M1]   1Ø7:      mov     al,Ø
[6V1]   1Ø8:      out     dx,al
[532]   1Ø9:      jc      exit14h
[ØØØ]   11Ø:
[1M1]   111:      sub     dx,4
[6L1]   112:      in      al,dx
[TV1]   113:      and     ah,1eh
[2B1]   114: exit14h:
[XD1]   115:      pop     dx
[VD1]   116:      pop     cx
[TD1]   117:      pop     bx
[X61]   118: end14h:
[GRØ]   119:      iret
[ØØØ]   12Ø:
[LGØ]   121: 11:
[RN1]   122:      and     ah,ah
[XA1]   123:      je      ahØ
[ØØØ]   124:
[SKØ]   125: ah1:
[SB1]   126:      inc     dx
[XF1]   127:      push    ax
[HL1]   128:      shr     cl,1
[HL1]   129:      shr     cl,1
[9H1]   13Ø:      and     cl,3
[P32]   131:      call    timing
[LA1]   132:      dec     dx
[891]   133:      jc      11Ø
[7A2]   134:      mov     cl,THRE
[P32]   135:      call    timing
[5HØ]   136: 11Ø:
[RD1]   137:      pop     ax
[7S1]   138:      xchg    al,ah
[6L1]   139:      in      al,dx
[7S1]   14Ø:      xchg    al,ah
[C91]   141:      jc      111
[2M1]   142:      sub     dx,5
[6V1]   143:      out     dx,al
[H53]   144:      jmp     short exit14h
[9HØ]   145: 111:
[ER1]   146:      or      ah,8Øh
[H53]   147:      jmp     short exit14h
[ØØØ]   148:
[PGØ]   149: 12:
[BK1]   15Ø:      cmp     ah,4
[SG1]   151:      jne     ah3
[ØØØ]   152:
[4LØ]   153: ah4:
[EV1]   154:      mov     bx,cx
[6T1]   155:      mov     ch,al
[MF2]   156:      jmp     short 112
[ØØØ]   157:
[ØØØ]   158:
```

Listing 1. »nmode.asm« zeigt einen neuen Schnittstellentreiber mit integriertem Hardware-Handshake

DOS INTERNATIONAL

Tips & Tricks gesucht!

Die DOS International ist eine Zeitschrift zum Mitmachen. Vor allem die Tips & Tricks-Rubriken leben von Ihren Beiträgen, Ihren Ideen und Programmen. Sie müssen kein Schriftsteller sein, damit ein Beitrag aus Ihrer Feder (beziehungsweise Ihrer Tastatur) veröffentlicht wird. Einschicken können Sie alles, was irgendwie mit der Programmierung und Bedienung des PC zusammenhängt. Das sind zum Beispiel wichtige Hinweise zu Anwendungs-Software, trickreiche Stapeldateien, interessante Basic-, Assembler-, C-, oder Pascal-Routinen, aber auch Tips zur PC-Hardware. Für jeden veröffentlichten Trick gibt es – je nach Länge und Qualität – ein angemessenes Honorar.

Bitte beachten Sie beim Vorbereiten einer Einsendung die folgenden Hinweise, damit der Beitrag in der Redaktion schnell und unproblematisch bearbeitet werden kann:

- Legen Sie eine Diskette bei mit allen zum Trick gehörigen Dateien (Quelltext, ausführbares Programm, Beschreibung).
- Das Anschreiben sollte in kurzen Stichpunkten Ihre Idee beschreiben.
- Diskette und Anschreiben versehen Sie bitte mit Ihrer genauen Adresse und mit dem Programmnamen.
- Sie ersparen uns viel Arbeit, wenn Sie Quelltexte auf 44 Zeichen pro Zeile formatieren. Steuerzeichen sollten nicht enthalten sein.
- Die Programmbeschreibung sollte als unformatierter ASCII-Text vorliegen.

Spätestens zwei Wochen, nachdem Ihre Einsendung bei uns eingegangen ist, erhalten Sie schriftlich Bescheid.

Unsere Adresse:

Redaktionsbüro
Everts & Hagedorn
Redaktion DOS International
Stichwort Tips & Tricks
Hans-Stießberger-Straße 3
8013 Haar/München

Wir freuen uns über jede Zuschrift.

```
[MKØ]  159: ahØ:
[6T1]  16Ø:     mov     ch,al
[EM1]  161:     mov     cl,5
[RS1]  162:     shr     al,cl
[DN1]  163:     add     al,al
[CV1]  164:     mov     bx,ax
[F95]  165:     mov     bx,word ptr cs:Baud[bx]
[DHØ]  166: 112:
[LA1]  167:     dec     dx
[LA1]  168:     dec     dx
[W72]  169:     mov     al,DLAB
[6V1]  17Ø:     out     dx,al
[ØM1]  171:     sub     dx,3
[FT1]  172:     mov     al,bl
[6V1]  173:     out     dx,al
[173]  174:     jmp     short wt_at1
[AC1]  175: wt_at1:
[SB1]  176:     inc     dx
[7T1]  177:     mov     al,bh
[6V1]  178:     out     dx,al
[273]  179:     jmp     short wt_at2
[DC1]  18Ø: wt_at2:
[SB1]  181:     inc     dx
[SB1]  182:     inc     dx
[8T1]  183:     mov     al,ch
[7W1]  184:     and     al,1fh
[6V1]  185:     out     dx,al
[SB1]  186:     inc     dx
[ØØØ]  187:
[5M1]  188:     mov     al,Ø
[6V1]  189:     out     dx,al
[373]  19Ø:     jmp     short wt_at3
[GC1]  191: wt_at3:
[1M1]  192:     sub     dx,4
[6L1]  193:     in      al,dx
[KH1]  194:     add     dx,4
[ØØØ]  195:
[ØØØ]  196:
[ØLØ]  197: ah3:
[6L1]  198:     in      al,dx
[473]  199:     jmp     short wt_at4
[LC1]  2ØØ: wt_at4:
[2T1]  2Ø1:     mov     ah,al
[SB1]  2Ø2:     inc     dx
[6L1]  2Ø3:     in      al,dx
[H53]  2Ø4:     jmp     short exit14h
[ØØØ]  2Ø5:
[TA1]  2Ø6: timing:
[MF1]  2Ø7:     push    ds
[VT1]  2Ø8:     sub     bx,bx
[2V1]  2Ø9:     mov     ds,bx
[CK4]  21Ø:     and     byte ptr [BrkFlg],7fh
[FU1]  211:     xchg    bx,cx
[3HØ]  212: t1:
[6L1]  213:     in      al,dx
[7U1]  214:     test    al,bl
[L91]  215:     jne     t3
[NN4]  216:     test    byte ptr [BrkFlg],8Øh
[G91]  217:     jne     t2
[CC1]  218:     loop    t1
[KK1]  219:     or      bh,bh
[Y31]  22Ø:     je      t1
[Z81]  221:     dec     bh
[D91]  222:     jne     t1
[6HØ]  223: t2:
[UR1]  224:     or      al,8Øh
[ENØ]  225:     stc
[9HØ]  226: t3:
[ED1]  227:     pop     ds
[2T1]  228:     mov     ah,al
[3PØ]  229:     ret
[ØØØ]  23Ø:
[SØ2]  231: int17h_14h:
[XK1]  232:     cmp     dl,4
[LN2]  233:     jae     end17h_14h
[ZF1]  234:     push    bx
[VT1]  235:     sub     bx,bx
[LT1]  236:     mov     bl,dl
[G36]  237:     test    byte ptr cs:Flags[bx],ParSer
[R31]  238:     je      14
[L95]  239:     mov     dl,byte ptr cs:Flags[bx]
[UR2]  24Ø:     and     dl,SerPort
[XF1]  241:     push    ax
[2A1]  242:     inc     ah
[9K1]  243:     cmp     ah,2
[Y32]  244:     jb      new_ah1
[342]  245:     ja      new_ah3
[TX1]  246:     mov     ah,8Øh
[8Z1]  247: exit17h_14h:
[TD1]  248:     pop     bx
[FT1]  249:     mov     al,bl
[371]  25Ø: end17h:
[TD1]  251:     pop     bx
[GY1]  252: end17h_14h:
[GRØ]  253:     iret
[ØØØ]  254:
[WGØ]  255: 14:
```

```
[RN1]  256:     and     ah,ah
[EF1]  257:     jne     142
[636]  258:     test    byte ptr cs:Flags[bx],ParWt
[Z91]  259:     je      142
[MF1]  26Ø:     push    ds
[VT1]  261:     sub     bx,bx
[2V1]  262:     mov     ds,bx
[CK4]  263:     and     byte ptr [BrkFlg],7fh
[ED1]  264:     pop     ds
[KHØ]  265: 141:
[ZXØ]  266:     pushf
[9L3]  267:     call    dword cs:int17h
[KM1]  268:     test    ah,1
[YT1]  269:     je      end17h
[MF1]  27Ø:     push    ds
[2V1]  271:     mov     ds,bx
[NN4]  272:     test    byte ptr [BrkFlg],8Øh
[ED1]  273:     pop     ds
[DZ1]  274:     jne     end17h
[8R1]  275:     sub     ah,ah
[NF2]  276:     jmp     short 141
[ØØØ]  277:
[PHØ]  278: 142:
[TD1]  279:     pop     bx
[ZL3]  28Ø:     jmp     dword cs:int17h
[ØØØ]  281:
[YB1]  282: new_ah1:
[AK1]  283:     int     14h
[HY1]  284:     test    ah,8Øh
[UHØ]  285: 143:
[UX1]  286:     mov     ah,9Øh
[HR2]  287:     je      exit17h_14h
[2M1]  288:     mov     ah,9
[TU3]  289:     jmp     short exit17h_14h
[ØØØ]  29Ø:
[6C1]  291: new_ah3:
[AK1]  292:     int     14h
[XU1]  293:     and     al,3Øh
[PF1]  294:     jne     144
[5Y1]  295:     mov     ah,29h
[TU3]  296:     jmp     short exit17h_14h
[YHØ]  297: 144:
[6Y1]  298:     xor     al,3Øh
[3A1]  299:     je      143
[KX1]  3ØØ:     mov     ah,1Øh
[TU3]  3Ø1:     jmp     short exit17h_14h
[ØØØ]  3Ø2:
[Z52]  3Ø3: endresidènt:
[ØØØ]  3Ø4:
[992]  3Ø5: esc     equ     27
[842]  3Ø6: cr      equ     13
[D32]  3Ø7: lf      equ     1Ø
[E52]  3Ø8: bell    equ     7
[ØØØ]  3Ø9:
[S74]  31Ø:         db      'NMODE v1.4 (c) 1989 MSB'
[ØØØ]  311:
[BH4]  312: SerDrv  db      'aux','com','ser'
[RH4]  313: ParDrv  db      'prn','lpt','par'
[ØØØ]  314:
[PV3]  315: ShortBd db      11,15,3Ø,6Ø,12
[9H2]  316:         db      24,48,96,19,38
[ØØØ]  317:
[D92]  318: ResFlg  db      -1
[Y82]  319: Command db      Ø
[TØ2]  32Ø: Port1   db      Ø
[VØ2]  321: Port2   db      Ø
[722]  322: Param1  db      Ø
[A22]  323: Param2  db      Ø
[A93]  324: Param3  db      ØØØ11Ø1Øxb
[ØØØ]  325:
[FH3]  326: ;Die Fehlermeldungen:
[ØØØ]  327:
[WH5]  328: mess_bp db      cr,lf,'Ungültiger '
[XT4]  329:         db      'Parameter',bell,cr,lf,'$'
[PT6]  33Ø: mess_bb db      cr,lf,'Ungültige Übertra'
[4X4]  331:         db      'gungsrate',bell,cr,lf,'$'
[9P6]  332: mess_rs db      cr,lf,'Residenter Teil von'
[GW4]  333:         db      ' NMODE geladen.',cr,lf,'$'
[376]  334: mess_ce db      cr,lf,'Serielle Schnitt'
[SX4]  335:         db      'stelle nicht vorhanden'
[GW2]  336:         db      bell,cr,lf,'$'
[696]  337: mess_pe db      cr,lf,'Parallele Schnitt'
[SX4]  338:         db      'stelle nicht vorhanden'
[GW2]  339:         db      bell,cr,lf,'$'
[3M5]  34Ø: mess_pf db      cr,lf,'Druckerfehler'
[GW2]  341:         db      bell,cr,lf,'$'
[ØØØ]  342:
[VD3]  343: ComRTM  db      lf,'- COM'
[SN2]  344: ComPar1 db      '1: '
[D43]  345: ComPar2 db      '1152ØØ,'
[XL2]  346: ComPar3 db      'E,'
[RK2]  347: ComPar4 db      '7,'
[GP3]  348: ComPar5 db      '1',cr,lf,'$'
[ØØØ]  349:
[3L3]  35Ø: LptRTM1 db      lf,'- LPT'
[214]  351: LptPar1 db      '1: auf COM'
[836]  352: LptPar2 db      '1: umgeleitet.',cr,lf,'$'
```

```
[ØØØ]  353:
[633]  354: LptRTM2 db      lf,'- '
[VW5]  355: LptPar3 db      ' 8Ø Zeichen pro Zeile.'
[YH1]  356:         db      cr,'$'
[ØØØ]  357:
[W53]  358: LptRTM3 db      lf,'-   '
[B76]  359: LptPar4 db      '6 Zeilen pro Zoll.',cr,'$'
[ØØØ]  36Ø:
[FL3]  361: LptRTM4 db      lf,'- LPT'
[SN5]  362: LptPar5 db      '1: nicht umgeleitet.'
[4Y1]  363:         db      cr,lf,'$'
[ØØØ]  364:
[ZP3]  365: WtRTM1  db      lf,'Keine '
[736]  366: WtRTM2  db      'Wiederholversuche bei '
[MR5]  367:         db      'Zeitüberschreitung durch '
[XT4]  368:         db      'Ausgabefehler.',cr,lf,'$'
[ØØØ]  369:
[RF2]  37Ø: set8Ø   db      1,12h
[WH2]  371: set132  db      1,Øfh
[B93]  372: set6pi  db      2,esc,32h
[B93]  373: set8pi  db      2,esc,3Øh
[ØØØ]  374:
[FG1]  375: calcbaud:
[BT2]  376:     mov     cx,BaudCnt
[GS1]  377:     and     dx,dx
[S91]  378:     jne     x2
[UN1]  379:     and     bh,bh
[N91]  38Ø:     jne     x1
[FT1]  381:     mov     al,bl
[NK1]  382:     cmp     al,2
[TLØ]  383:     cmc
[1F1]  384:     jnc     15Ø
[SZ3]  385:     mov     si,offset ShortBd
[4V1]  386:     mov     dx,si
[SS1]  387:     xchg    di,si
[NY1]  388:     repne   scasb
[SS1]  389:     xchg    di,si
[N91]  39Ø:     jne     x1
[KT1]  391:     sub     si,dx
[MP1]  392:     add     si,si
[KZ4]  393:     mov     ax,word ptr Baud[si-2]
[2E2]  394:     jmp     short x3
[BHØ]  395: x1:
[VR2]  396:     mov     dx,MxBaudH
[1S2]  397:     mov     ax,MxBaudL
[7D1]  398:     div     bx
[2E2]  399:     jmp     short x3
[EHØ]  4ØØ: x2:
[HP2]  4Ø1:     cmp     bx,MxBaudL
[E91]  4Ø2:     ja      15Ø
[DV1]  4Ø3:     mov     ax,dx
[HHØ]  4Ø4: x3:
[154]  4Ø5:     mov     word ptr Param1,ax
[CV1]  4Ø6:     mov     bx,ax
[DA1]  4Ø7:     dec     ax
[SA1]  4Ø8:     je      x33
[VR2]  4Ø9:     mov     dx,MxBaudH
[1S2]  41Ø:     mov     ax,MxBaudL
[7D1]  411:     div     bx
[GN1]  412:     mov     bx,5
[NU3]  413:     mov     si,offset ComPar2
[AR1]  414:     mov     cl,1Ø
[7KØ]  415: x31:
[1U1]  416:     sub     dx,dx
[9D1]  417:     div     cx
[PP1]  418:     add     dl,'Ø'
[8K4]  419:     mov     byte ptr [si+bx],dl
[FA1]  42Ø:     dec     bx
[7S1]  421:     and     ax,ax
[ZF1]  422:     jne     x31
[BKØ]  423: x32:
[8B4]  424:     mov     byte ptr [si+bx],' '
[FA1]  425:     dec     bx
[VH1]  426:     jns     x32
[MB1]  427:     inc     bx
[FKØ]  428: x33:
[ENØ]  429:     stc
[HHØ]  43Ø: 15Ø:
[3PØ]  431:     ret
[ØØØ]  432:
[5E1]  433: getchar:
[ZX1]  434:     mov     al,2Øh
[6T1]  435:     mov     ch,al
[2W1]  436:     repe    scasb
[691]  437:     dec     di
[3PØ]  438:     ret
[ØØØ]  439:
[TG1]  44Ø: getdrvr:
[SS1]  441:     xchg    di,si
[8R1]  442:     sub     ah,ah
[NHØ]  443: 151:
[DN1]  444:     mov     bx,2
[THØ]  445: 152:
[CVØ]  446:     lodsb
[MR1]  447:     or      al,2Øh
[AUØ]  448:     scasb
```

```
[NF1]  449:     jne     153
[FA1]  45Ø:     dec     bx
[9H1]  451:     jns     152
[3PØ]  452:     ret
[XHØ]  453: 153:
[3P1]  454:     add     di,bx
[1R1]  455:     add     si,bx
[HL1]  456:     sub     si,3
[2A1]  457:     inc     ah
[AK1]  458:     cmp     ah,3
[N91]  459:     jb      151
[SS1]  46Ø:     xchg    di,si
[3PØ]  461:     ret
[ØØØ]  462:
[VG1]  463: getport:
[RN1]  464:     and     ah,ah
[AA1]  465:     je      154
[CVØ]  466:     lodsb
[1T1]  467:     sub     al,'1'
[CA1]  468:     jc      156
[RK1]  469:     cmp     al,4
[TF1]  47Ø:     jnc     156
[2T1]  471:     mov     ah,al
[1KØ]  472: 154:
[BU3]  473:     cmp     byte ptr [si],':'
[XF1]  474:     jne     155
[9B1]  475:     inc     si
[5KØ]  476: 155:
[SS1]  477:     xchg    di,si
[PLØ]  478:     clc
[3PØ]  479:     ret
[9KØ]  48Ø: 156:
[ENØ]  481:     stc
[3PØ]  482:     ret
[ØØØ]  483:
[BT1]  484: getnumber:
[GC2]  485:     call    getchar
[SS1]  486:     xchg    di,si
[1U1]  487:     sub     dx,dx
[ES1]  488:     mov     cx,1Ø
[CVØ]  489:     lodsb
[ZS1]  49Ø:     sub     al,'Ø'
[XE1]  491:     jbe     156
[YR1]  492:     cmp     al,cl
[UE1]  493:     jae     156
[8R1]  494:     sub     ah,ah
[DKØ]  495: 157:
[CU1]  496:     xchg    ax,bx
[CVØ]  497:     lodsb
[ZS1]  498:     sub     al,'Ø'
[HA1]  499:     jb      158
[YR1]  5ØØ:     cmp     al,cl
[EA1]  5Ø1:     ja      158
[8R1]  5Ø2:     sub     ah,ah
[DU1]  5Ø3:     xchg    bx,ax
[RD1]  5Ø4:     mul     cx
[AR1]  5Ø5:     add     ax,bx
[AH1]  5Ø6:     adc     dx,Ø
[PA1]  5Ø7:     je      157
[CV1]  5Ø8:     mov     bx,ax
[9B1]  5Ø9:     inc     si
[HKØ]  51Ø: 158:
[4A1]  511:     dec     si
[SS1]  512:     xchg    di,si
[GC2]  513:     call    getchar
[PLØ]  514:     clc
[3PØ]  515:     ret
[ØØØ]  516:
[TP1]  517: checkparm:
[8R1]  518:     sub     ah,ah
[1S3]  519:     cmp     byte ptr [di],','
[CF1]  52Ø:     jne     16Ø
[BA1]  521:     inc     di
[GC2]  522:     call    getchar
[1S3]  523:     cmp     byte ptr [di],','
[1A1]  524:     je      161
[MV3]  525:     mov     ah,byte ptr [di]
[BA1]  526:     inc     di
[GC2]  527:     call    getchar
[BP1]  528:     and     al,al
[3PØ]  529:     ret
[MHØ]  53Ø: 16Ø:
[ENØ]  531:     stc
[SHØ]  532: 161:
[3PØ]  533:     ret
[ØØØ]  534:
[PM1]  535: error_bb:
[ZØ4]  536:     mov     dx,offset mess_bb
[KØ4]  537:     jmp     short exit_error
[ØØØ]  538:
[4N1]  539: error_bp:
[V14]  54Ø:     mov     dx,offset mess_bp
[ØØØ]  541:
[D52]  542: exit_error:
[2M1]  543:     mov     ah,9
```

Listing 1. »nmode.asm« zeigt einen neuen Schnittstellentreiber mit integriertem Hardware-Handshake

```
[3K1]  544:      int     21h
[8R1]  545:      sub     ah,ah
[3K1]  546:      int     21h
[ØØØ]  547:
[P11]  548: parse:
[ULØ]  549:      cld
[4Y1]  55Ø:      mov     di,81h
[GC2]  551:      call    getchar
[NU3]  552:      mov     si,offset SerDrv
[4F2]  553:      call    getdrvr
[482]  554:      je      chk_lpt
[6F2]  555:      call    getport
[DB2]  556:      jc      error_bp
[FX3]  557:      mov     byte ptr Port1,ah
[ØØØ]  558:
[XA4]  559:      add     byte ptr ComPar1,ah
[ØØØ]  56Ø:
[2M2]  561:      call    getnumber
[ZA2]  562:      jc      error_bb
[ØD2]  563:      call    calcbaud
[CG2]  564:      jnc     error_bb
[ØØØ]  565:
[SF4]  566: ;calcbaud setzt BH auf Null
[ØØØ]  567:
[GH2]  568:      call    checkparm
[762]  569:      jc      end_com
[G42]  57Ø:      and     ah,Ødfh
[FU1]  571:      cmp     ah,'N'
[Y91]  572:      jne     x4
[CR2]  573:      mov     bh,ØØØ11ØØØxb
[5H2]  574:      jmp     short x41
[MHØ]  575: x4:
[HU1]  576:      cmp     ah,'O'
[1A1]  577:      jne     x5
[8R2]  578:      mov     bh,ØØØØ1ØØØxb
[AKØ]  579: x41:
[ØF4]  58Ø:      mov     byte ptr ComPar3,ah
[RHØ]  581: x5:
[GH2]  582:      call    checkparm
[762]  583:      jc      end_com
[3T1]  584:      cmp     ah,'8'
[4A1]  585:      jne     x6
[4A1]  586:      inc     bh
[ØØØ]  587:
[FU3]  588:      inc     byte ptr ComPar4
[UHØ]  589: x6:
[GH2]  59Ø:      call    checkparm
[762]  591:      jc      end_com
[RS1]  592:      cmp     ah,'2'
[7A1]  593:      jne     x7
[YG2]  594:      or      bh,ØØØØØ1ØØxb
[ØØØ]  595:
[GU3]  596:      inc     byte ptr ComPar5
[XHØ]  597: x7:
[GH2]  598:      call    checkparm
[762]  599:      jc      end_com
[8R1]  6ØØ:      or      ah,2Øh
[LU1]  6Ø1:      cmp     ah,'p'
[WB2]  6Ø2:      jne     end_com
[GR1]  6Ø3:      or      bh,8Øh
[4E1]  6Ø4: end_com:
[Y34]  6Ø5:      xor     byte ptr Param3,bh
[DE2]  6Ø6:      jmp     install
[ØØØ]  6Ø7:
[VF1]  6Ø8: chk_lpt:
[9U3]  6Ø9:      inc     byte ptr Command
[5U3]  61Ø:      mov     si,offset ParDrv
[4F2]  611:      call    getdrvr
[HM2]  612:      je      to_err_bp
[6F2]  613:      call    getport
[BM2]  614:      jc      to_err_bp
[FX3]  615:      mov     byte ptr Port1,ah
[ØØØ]  616:
[3C4]  617:      add     byte ptr LptPar1,ah
[7C4]  618:      add     byte ptr LptPar5,ah
[ØØØ]  619:
[GC2]  62Ø:      call    getchar
[KS3]  621:      cmp     byte ptr [di],'='
[VR1]  622:      jne     lpt1
[BA1]  623:      inc     di
[GC2]  624:      call    getchar
[NU3]  625:      mov     si,offset SerDrv
[4F2]  626:      call    getdrvr
[HM2]  627:      je      to_err_bp
[6F2]  628:      call    getport
[BM2]  629:      jc      to_err_bp
[KX3]  63Ø:      mov     byte ptr Port2,ah
[ØØØ]  631:
[4C4]  632:      add     byte ptr LptPar2,ah
[ØØØ]  633:
[B93]  634:      jmp     short install
[ØØØ]  635:
[WZ1]  636: to_err_bp:
[BK2]  637:      jmp     error_bp
[ØØØ]  638:
[XVØ]  639: lpt1:
```

```
[9U3]  64Ø:      inc     byte ptr Command
[2M2]  641:      call    getnumber
[N41]  642:      jc      x8
[GS1]  643:      and     dx,dx
[ØU2]  644:      jne     to_err_bp
[PV1]  645:      cmp     bx,132
[7Y1]  646:      mov     dl,4Øh
[AA1]  647:      jne     x8
[SB1]  648:      inc     dx
[ØØØ]  649:
[6F4]  65Ø:      mov     word ptr LptPar3,'31'
[7L4]  651:      mov     byte ptr LptPar3+2,'2'
[ØKØ]  652: x8:
[GH2]  653:      call    checkparm
[992]  654:      jc      end_lpt1
[6A1]  655:      je      x1Ø
[4U1]  656:      add     dl,2Øh
[3T1]  657:      cmp     ah,'8'
[NF1]  658:      jne     x1Ø
[CG1]  659:      add     dl,2
[ØØØ]  66Ø:
[7G4]  661:      mov     byte ptr LptPar4,ah
[XHØ]  662: x1Ø:
[GH2]  663:      call    checkparm
[992]  664:      jc      end_lpt1
[8R1]  665:      or      ah,2Øh
[LU1]  666:      cmp     ah,'p'
[YE2]  667:      jne     end_lpt1
[ØS1]  668:      or      dl,8Øh
[ØL1]  669: end_lpt1:
[Ø44]  67Ø:      mov     byte ptr Param1,dl
[ØØØ]  671:
[CF1]  672: install:
[162]  673:      mov     ax,3514h
[3K1]  674:      int     21h
[4U1]  675:      mov     di,bx
[2V1]  676:      mov     si,bx
[KN1]  677:      mov     cx,5
[KNØ]  678:      std
[WX1]  679:      repe    cmpsb
[GH1]  68Ø:      je      doit
[ØØØ]  681:
[7S3]  682:      inc     byte ptr ResFlg
[D62]  683:      mov     ax,3517h
[3K1]  684:      int     21h
[Y54]  685:      mov     word ptr int17h,bx
[ØC4]  686:      mov     word ptr int17h+2,es
[C62]  687:      mov     ax,2517h
[Z74]  688:      mov     dx,offset int17h_14h
[EMØ]  689:      cli
[3K1]  69Ø:      int     21h
[6PØ]  691:      sti
[Ø62]  692:      mov     ax,2514h
[AM3]  693:      mov     dx,offset int14h
[EMØ]  694:      cli
[3K1]  695:      int     21h
[6PØ]  696:      sti
[H24]  697:      mov     dx,offset mess_rs
[2M1]  698:      mov     ah,9
[3K1]  699:      int     21h
[MF1]  7ØØ:      push    ds
[GD1]  7Ø1:      pop     es
[ØØØ]  7Ø2:
[1VØ]  7Ø3: doit:
[AØ4]  7Ø4:      mov     bl,byte ptr Port1
[BR1]  7Ø5:      sub     bh,bh
[HV1]  7Ø6:      mov     dx,bx
[ST1]  7Ø7:      sub     ax,ax
[4T3]  7Ø8:      dec     byte ptr Command
[1V1]  7Ø9:      mov     ds,ax
[382]  71Ø:      jns     do_lpt
[1D1]  711: do_com:
[2V1]  712:      mov     si,bx
[6A5]  713:      test    word ptr ComAdr[bx+si],-1
[KF1]  714:      push    cs
[ED1]  715:      pop     ds
[H82]  716:      je      com_err
[995]  717:      and     byte ptr es:Flags[bx],7fh
[W44]  718:      mov     al,byte ptr Param3
[BP1]  719:      and     al,al
[HH1]  72Ø:      jns     x11
[DW1]  721:      and     al,7fh
[TW5]  722:      or      byte ptr es:Flags[bx],SerWt
[1KØ]  723: x11:
[N54]  724:      mov     cx,word ptr Param1
[5G1]  725: set_com:
[XL1]  726:      mov     ah,4
[AK1]  727:      int     14h
[DU3]  728:      mov     dx,offset ComRTM
[6H2]  729:      jmp     short x13
[ØØØ]  73Ø:
[1E1]  731: do_lpt:
[2V1]  732:      mov     si,bx
[U95]  733:      test    word ptr ParAdr[bx+si],-1
[N82]  734:      je      par_err
[SD4]  735:      dec     byte ptr cs:Command
```

```
[Z92]  736:      je      set_lpt
[5L4]  737:      mov     ax,word ptr cs:Port2
[DU1]  738:      xchg    bx,ax
[2V1]  739:      mov     si,bx
[6A5]  74Ø:      test    word ptr ComAdr[bx+si],-1
[KF1]  741:      push    cs
[ED1]  742:      pop     ds
[H82]  743:      je      com_err
[CU1]  744:      xchg    ax,bx
[VE5]  745:      and     byte ptr es:Flags[bx],Øfch
[HE2]  746:      add     al,ParSer
[Z25]  747:      or      byte ptr es:Flags[bx],al
[MR2]  748:      and     al,SerPort
[DU1]  749:      xchg    bx,ax
[2Z3]  75Ø:      mov     dx,offset LptRTM1
[9KØ]  751: x13:
[W36]  752:      test    byte ptr es:Flags[bx],SerWt
[CB3]  753:      jmp     short do_exit
[ØØØ]  754:
[8G1]  755: com_err:
[614]  756:      mov     dx,offset mess_ce
[N53]  757:      jmp     short do_err
[ØØØ]  758:
[CG1]  759: par_err:
[KF1]  76Ø:      push    cs
[ED1]  761:      pop     ds
[L14]  762:      mov     dx,offset mess_pe
[N53]  763:      jmp     short do_err
[ØØØ]  764:
[PH1]  765: set_lpt:
[KF1]  766:      push    cs
[ED1]  767:      pop     ds
[ØØØ]  768:
[H95]  769:      and     byte ptr es:Flags[bx],9fh
[CA4]  77Ø:      test    byte ptr Param1,8Øh
[PA1]  771:      je      x14
[9W5]  772:      or      byte ptr es:Flags[bx],ParWt
[DKØ]  773: x14:
[7A4]  774:      test    byte ptr Param1,41h
[YA1]  775:      je      x16
[ØM3]  776:      mov     si,offset set132
[FG1]  777:      jpe     x15
[RK3]  778:      mov     si,offset set8Ø
[HKØ]  779: x15:
[UH2]  78Ø:      call    send_prn
[7F2]  781:      jne     prn_err
[3G1]  782:      push    dx
[4Z3]  783:      mov     dx,offset LptRTM2
[2M1]  784:      mov     ah,9
[3K1]  785:      int     21h
[XD1]  786:      pop     dx
[NKØ]  787: x16:
[6A4]  788:      test    byte ptr Param1,22h
[6B1]  789:      je      x18
[ØT3]  79Ø:      mov     si,offset set8pi
[RG1]  791:      jpe     x17
[US3]  792:      mov     si,offset set6pi
[TKØ]  793: x17:
[UH2]  794:      call    send_prn
[7F2]  795:      jne     prn_err
[6Z3]  796:      mov     dx,offset LptRTM3
[2M1]  797:      mov     ah,9
[3K1]  798:      int     21h
[XKØ]  799: x18:
[VL1]  8ØØ:      mov     ah,2
[LT1]  8Ø1:      mov     dl,lf
[3K1]  8Ø2:      int     21h
[8Z3]  8Ø3:      mov     dx,offset LptRTM4
[C36]  8Ø4:      test    byte ptr es:Flags[bx],ParWt
[CB3]  8Ø5:      jmp     short do_exit
[ØØØ]  8Ø6:
[FH1]  8Ø7: prn_err:
[N14]  8Ø8:      mov     dx,offset mess_pf
[N53]  8Ø9:      jmp     short do_err
[ØØØ]  81Ø:
[BK1]  811: do_exit:
[3G1]  812:      push    dx
[UV3]  813:      mov     dx,offset WtRTM1
[2C1]  814:      je      x99
[VL1]  815:      mov     ah,2
[LT1]  816:      mov     dl,lf
[3K1]  817:      int     21h
[VV3]  818:      mov     dx,offset WtRTM2
[TLØ]  819: x99:
[2M1]  82Ø:      mov     ah,9
[3K1]  821:      int     21h
[XD1]  822:      pop     dx
[SD1]  823: do_err:
[2M1]  824:      mov     ah,9
[3K1]  825:      int     21h
[ØØØ]  826:
[YVØ]  827: exit:
[F44]  828:      test    byte ptr ResFlg,-1
[8B2]  829:      je      exit_res
[8R1]  83Ø:      sub     ah,ah
[3K1]  831:      int     21h
```

```
[TM1]  832: exit_res:
[ØØØ]  833:
[BZ3]  834:      mov     es,word ptr [2ch]
[7Y1]  835:      mov     ah,49h
[3K1]  836:      int     21h
[ML4]  837:      mov     dx,offset endresident
[NK1]  838:      int     27h
[ØØØ]  839:
[9N1]  84Ø: send_prn:
[ULØ]  841:      cld
[CVØ]  842:      lodsb
[KT1]  843:      mov     cl,al
[ER1]  844:      sub     ch,ch
[NLØ]  845: x98:
[8R1]  846:      sub     ah,ah
[CVØ]  847:      lodsb
[ZXØ]  848:      pushf
[HL3]  849:      call    dword es:int17h
[872]  85Ø:      test    ah,Øf9h
[PP1]  851:      loope   x98
[3PØ]  852:      ret
[ØØØ]  853:
[ZP1]  854: code    ends
[4W1]  855:         end     start
```

Listing 1. »nmode.asm« zeigt einen neuen Schnittstellentreiber mit integriertem Hardware-Handshake

Name	com2bin.c
Sprache	Turbo C, Quick C, Microsoft C
Grafikkarte	jede
Besonderheiten	keine

```
[PØ3]    1: /* Utility Com2Bin:
[7Y5]    2:    (c) Redaktionsbüro Everts & Hagedorn
[387]    3:    Aufruf: COM2BIN dest [Pport] [Ttimeout]
[ØØØ]    4:
[YR5]    5:    Sendet die binäre Information von
[UØ6]    6:    COMport an die Datei dest. Abbruch der
[247]    7:    Übertragung durch [Ctrl-Break] oder nach
[8S4]    8:    Ablauf der vorgegebenen Zeit.
[GY6]    9:    Handshake-Protokoll muß vorher durch
[544]   1Ø:    NMODE aktiviert werden.
[ØØØ]   11:
[WY1]   12:    Sprache: C */
[ØØØ]   13:
[U83]   14: #include <stdlib.h>
[L13]   15: #include <stdio.h>
[9L2]   16: #include <dos.h>
[ØØØ]   17:
[Z22]   18: union REGS r;
[9D3]   19: int file,param,pval;
[876]   2Ø: char port,oldtime,newtime,buffer[1];
[ØØØ]   21:
[416]   22: exiterror(errmsg) char errmsg[]; {
[EB4]   23:   puts(errmsg); puts(".\n");
[F61]   24:   exit(2);
[U7Ø]   25: }
[ØØØ]   26:
[DR6]   27: main(argc,argv) int argc; char *argv[]; {
[ØØØ]   28:
[SE6]   29: /* Vorgabe: COM1 und Wiederholversuche
[3C5]   3Ø:    innerhalb 1 sec (PC/XT 4.77 MHz) */
[ØØØ]   31:
[DR2]   32:   port=newtime=1;
[ØØØ]   33:
[BU5]   34: /* wenigstens Dateiname angegeben? */
[ØØØ]   35:
[NF7]   36:   if(argc<2) exiterror("Ungültiger Paramete
        r");
[ØØØ]   37:
[UG6]   38: /* Datei erzeugen (eventuell vorhandene
[K96]   39:    wird gelöscht!) und Abbruch im Fehler-
[521]   4Ø:    fall */
[ØØØ]   41:
[A4C]   42:   if((file=creat(argv[1]))==-1) exiterror("
        Datei kann nicht erstellt werden");
[ØØØ]   43:
[GG6]   44: /* Die weiteren Parameter werden auf die
[326]   45:    Muster Pnummer und Tnummer hin unter-
[UC6]   46:    sucht und die entsprechenden Variablen
```

Listing 2. »com2bin.c« überträgt binäre Informationen von der seriellen Schnittstelle an eine Datei

```
[UA1]   47:    belegt */
[ØØØ]   48:
[4A1]   49:   param=3;
[P63]   5Ø:   while(argc>=param) {
[8P4]   51:     pval=atoi(argv[--param]+1);
[Z54]   52:     pval=(pval&ØxFFØØ)?Ø:pval;
[HH4]   53:     if((*argv[param]|Øx2Ø)=='p')
[NV1]   54:       port=pval;
[ZRØ]   55:     else
[XH4]   56:     if((*argv[param]|Øx2Ø)=='t')
[DA2]   57:       newtime=pval;
[3D1]   58:     param+=2;
[U7Ø]   59:   }
[ØØØ]   6Ø:
[6Ø6]   61: /* Existiert die angegebene serielle
[GU2]   62:    Schnittstelle? */
[ØØØ]   63:
[XP6]   64:   if((--port>3)||(peekw(Øx4Ø,port<<1)==Ø))
        {
[2X1]   65:     close(file);
[B36]   66:     exiterror("COM-Port nicht gefunden");
[U7Ø]   67:   }
[2T4]   68:   oldtime=peekb(Øx4Ø,Øx7C+port);
[ØØØ]   69:
[683]   7Ø: /* Der Datentransfer:
[5D6]   71:    Es wird ein Zeichen von der seriellen
[326]   72:    Schnittstelle gelesen. Abbruch, wenn
[1W5]   73:    innerhalb der angegebenen Zeit kein
[6W5]   74:    Zeichen gelesen werden konnte oder
[HØ5]   75:    [CTRL-BREAK] gedrückt wurde.
[MY6]   76:    Ein Datenübertragungsfehler wird ange-
[Ø11]   77:    zeigt.
[T96]   78:    Das gelesene Zeichen wird in die Datei
[PG6]   79:    geschrieben. Sollte dies fehlschlagen,
[N45]   8Ø:    erfolgt auch hier ein Abbruch */
[ØØØ]   81:
[PM1]   82:   while(1) {
[KB1]   83:     r.h.ah=2;
[F52]   84:     r.x.dx=port;
[BU4]   85:     pokeb(Øx4Ø,Øx7C+port,newtime);
[9F2]   86:     int86(Øx14,&r,&r);
[RU4]   87:     pokeb(Øx4Ø,Øx7C+port,oldtime);
[C34]   88:     if(peekb(Øx4Ø,Øx71)&Øx8Ø) {
[2X1]   89:       close(file);
[C61]   9Ø:       exit(1);
[U7Ø]   91:     }
[ZRØ]   92:     else
[N98]   93:     if(r.h.ah&Øx1E) puts("Datenübertragungs
        fehler!\n");
[ZRØ]   94:     else
[RA2]   95:     if(r.h.ah&Øx8Ø) {
[2X1]   96:       close(file);
[YK9]   97:       exiterror("Empfang abgebrochen nach Z
        eitüberschreitung");
[U7Ø]   98:     }
[G51]   99:     else {
[4Y2]  1ØØ:       buffer[Ø]=r.h.al;
[VM4]  1Ø1:       if(write(file,buffer,1)==-1) {
[2X1]  1Ø2:         close(file);
[XH4]  1Ø3:         exiterror("Schreibfehler");
[U7Ø]  1Ø4:       }
[U7Ø]  1Ø5:     }
[U7Ø]  1Ø6:   }
[U7Ø]  1Ø7: }
```

Listing 2. »com2bin.c« überträgt binäre Informationen von der seriellen Schnittstelle an eine Datei

Name	confcom.c
Sprache	Turbo C, Quick C, Microsoft C
Grafikkarte	jede
Besonderheiten	keine

```
[L33]    1: /* Utility ConfCom:
[7Y5]    2:    (c) Redaktionsbüro Everts & Hagedorn
[ØØØ]    3:
[ZG6]    4:    Programm zur Festlegung des Hardware-
[5V6]    5:    Protokolls der seriellen Schnittstelle.
[PR6]    6:    Das Programm fragt die notwendigen Daten
[NS6]    7:    interaktiv ab und erlaubt für jeden Port
[9X3]    8:    ein eigenes Protokoll.
[ØØØ]    9:
[HD3]   1Ø:    Aufruf durch: CONFCOM
[ØØØ]   11:
[WY1]   12:    Sprache: C */
[ØØØ]   13:
[U83]   14: #include <stdlib.h>
[L13]   15: #include <stdio.h>
[ØØØ]   16:
[5Y6]   17: /* Der DeSmet C-Compiler kennt die Funktion
[RG6]   18:    ci(), welche ein Zeichen von stdin liest
[4E6]   19:    und dieses als (unsigned) char zurück-
[DT5]   2Ø:    gibt. Borlands Turbo-C hat schon eine
[5N6]   21:    mit getch() bezeichnete Funktion (evtl.
[826]   22:    in einer oben nicht genannten Header-
[U16]   23:    Datei), deren Einbindung anstelle des
[3A5]   24:    folgenen define stehen sollte. */
[ØØØ]   25:
[HK2]   26: #define getch ci
[ØØØ]   27:
[B46]   28: /* Die union ADDR erlaubt den einfachen
[SW5]   29:    Zugriff auf die beiden 16-Bit Worte
[233]   3Ø:    eines Langworts */
[ØØØ]   31:
[YS4]   32: struct SEGS { int ofs,seg; };
[TS6]   33: union ADDR { struct SEGS s; long adr; };
[893]   34: char ident[]="NCOM";
[G37]   35: char *term[]={ "?","DTR","RTS","RTS/DTR" };
[FZ6]   36: char *send[]={ "?","CTS","DSR","CTS/DSR" };
[ØØØ]   37:
[N41]   38: main() {
[RC3]   39:   char port,rlin,slin;
[F11]   4Ø:   int i;
[MK2]   41:   union ADDR ivec;
[746]   42:   ivec.adr=peekl(Ø,Øx14*4); /* getintvec */
[ØØØ]   43:
[3E6]   44: /* die folgende Erkennung basiert auf der
[SE6]   45:    Tatsache, daß der NMODE-Treiber an der
[176]   46:    Bytefolge 'NCOM' erkannt werden kann,
[6N6]   47:    welche direkt unterhalb des neuen INT14-
[773]   48:    Einsprungs steht */
[ØØØ]   49:
[LM9]   5Ø:   for(i=4;(ident[--i]==peekb(ivec.s.seg,--i
        vec.s.ofs))&&i>=Ø;);
[UG1]   51:   if(i>=Ø) {
[4K6]   52:     puts("NMODE nicht installiert!\ØØ7\n");
[C61]   53:     exit(1);
[U7Ø]   54:   }
[ØØØ]   55:
[2D6]   56: /* die Differenz von 24 Bytes ergibt sich
[2S6]   57:    aus dem 'offset HandShk' und dem 'offset
[HP4]   58:    Ident' im .COM-File NMODE (vgl.
[BW1]   59:    NMODE.ASM) */
[ØØØ]   6Ø:
[9D2]   61:   ivec.s.ofs-=24;
[PM1]   62:   while(1) {
[Ø88]   63:     puts("\n\nHandshakeprotokoll eingestell
        t:\n");
[ØØØ]   64:
[CM6]   65: /* Ausgabe der momentan gewählten Proto-
[AD6]   66:    kolle der seriellen Ports 1..4. Hierbei
[D16]   67:    und im folgenen ist unerheblich, ob
[BH6]   68:    diese Ports überhaupt vorhanden sind.
[ZR6]   69:    Zugriffe auf nicht existente Ports wer-
[TB6]   7Ø:    den von NMODE und COM2BIN abgelehnt. */
[ØØØ]   71:
[ZS2]   72:     for(i=1;i<5;i++) {
[2L4]   73:       puts("\nCOM"); putchar(i+'Ø');
[BN1]   74:       puts(": ");
[NH7]   75:       puts(term[peekb(ivec.s.seg,ivec.s.ofs
        +i)&3]);
[8M2]   76:       puts("\t<-->\t");
[L18]   77:       puts(send[(peekb(ivec.s.seg,ivec.s.of
        s+i)>>2)&3]);
[U7Ø]   78:     }
[LS7]   79:     puts("\n\nZu änderner Port (1..4) [Ø =
        ENDE]? ");
[BS6]   8Ø:     do ; while(((port=getch())<'Ø')||(port>
        '4'));
[A63]   81:     if(port=='Ø') exit(Ø);
[1H2]   82:     putchar(port);
[4Ø8]   83:     puts("\n\nDTE-Leitung (1:DTR 2:RTS 3:RT
        S/DTR)? ");
[HP6]   84:     do ; while(((rlin=getch())<'1')||(rlin>
        '3'));
[UL3]   85:     puts(term[rlin-='Ø']);
[ZE7]   86:     puts("\nDCE-Leitung (1:CTS 2:DSR 3:CTS/
        DSR)? ");
[SP6]   87:     do ; while(((slin=getch())<'1')||(slin>
        '3'));
[5L3]   88:     puts(send[slin-='Ø']);
[ØØØ]   89:
[HT6]   9Ø: /* Da die gewünschte Änderung direkt im
[TX5]   91:    resident geladenen Treiber von NMODE
[NP5]   92:    vorgenommen wird, sollte man beim
[PK6]   93:    Abtippen natürlich besondere Vorsicht
[1U6]   94:    walten lassen. Zur Eingabekontrolle wird
[886]   95:    diese Änderung nochmals angezeigt. */
[ØØØ]   96:
[7W8]   97:     pokeb(ivec.s.seg,ivec.s.ofs+(port-'Ø'),
        ((slin<<2)+rlin));
[U7Ø]   98:   }
[U7Ø]   99: }
```

Listing 3. Mit »confcom.c« entscheiden Sie, welche Leitungen für das Handshake-Verfahren eingesetzt werden

Nützlicher Helfer im Hintergrund: Der TSR-Dateimanager

Wer wollte während der Ausführung eines Programms nicht schon einmal auf MS-DOS-Funktionen zugreifen, die aber dann von dem Anwendungsprogramm (natürlich) nicht unterstützt wurden. Der TSR-Dateimanager »Dmanager« schafft hier Abhilfe. Er stellt Ihnen die üblichen Dateiverwaltungsfunktionen (Kopieren, Löschen, Umbenennen und so weiter) zur Verfügung und bietet darüber hinaus nützliche Erweiterungen.

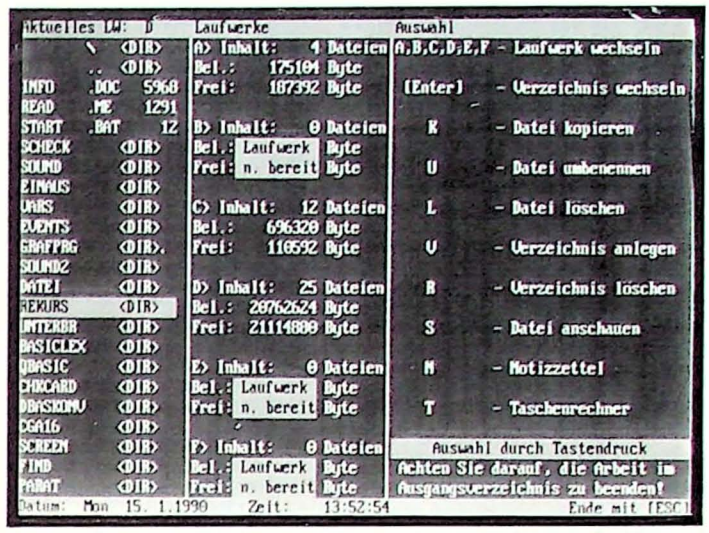


Bild. Nützliche MS-DOS-Funktionen stets im Zugriff – der »Dmanager« macht's möglich

Wohl jeder Anwender befand sich bereits einmal in einer Situation, die während des Ablaufs eines Programms einen Wechsel auf die MS-DOS-Betriebssystemebene notwendig machte. Wie das Leben so spielt, ließ die Anwendung dies jedoch nicht zu. Ein typischer Fall für den TSR-Dateimanager »Dmanager». Der Start eines neuen Programms innerhalb einer Dosshell ist mit ihm zwar prinzipiell nicht möglich, die üblichen Dateiverwaltungsfunktionen – wie Kopieren (»copy«), Umbenennen (»rename«), Löschen (»delete«), Anlegen (»mkdir«) und Löschen (»rmdir«) von Verzeichnissen, Auflisten (»dir«) des Inhalts eines Laufwerks beziehungsweise Verzeichnisses sowie Ausgabe (»type«) des Inhalts einer Datei auf dem Bildschirm und Wechsel des Laufwerks und des aktuellen Verzeichnisses (»chdir«) – stellt Dmanager aber zur Verfügung. Daneben enthält es noch einen einfachen »Taschenrechner« und eine Notizzettelfunktion, die beliebig lange Textdateien erzeugt.

Alle benutzten MS-DOS-Funktionen sind in »normalen« Turbo-Pascal-Programmen sehr einfach über den Aufruf des Kommandoprozessors »command.com« zu realisieren. Der Programmieraufwand ist mäßig, und es ist lediglich darauf zu achten, daß während der Laufzeit des Programms der Befehls-Interpreter »command.com« irgendwo zur Verfügung steht. Wenn Sie dieses Verfahren bei einem speicherresidenten Programm ausprobieren, merken Sie sofort, daß es nicht funktioniert. Das hat seinen Grund darin, daß sich aus einem speicherresidenten Programm keine Tochterprozesse über den Aufruf der Exec-Prozedur starten lassen.

Folglich sind Sie gezwungen, die gebräuchlichsten MS-DOS-Funktionen selbst zu programmieren. Das erweist sich allerdings als nicht besonders schwierig, denn Turbo Pascal stellt eine Reihe von Prozeduren zur Verfügung, die im Grunde den MS-DOS-Funktionen entsprechen. Der Programmieraufwand ist daher – abgesehen von einigen Fällen, die eine maschinennähere Interrupt-Programmierung erforderlich machen – nur noch für die entsprechende Benutzeroberfläche notwendig.

Die in einigen Programmteilen etwas umständlich anmutende Art der Programmierung resultiert aus dem Bestreben, den TSR-Dateimanager möglichst »wasserdicht« zu machen, das heißt, ihn gegen Bedienungsfehler abzusichern. Selbstverständlich ist die Sicherung nicht vollkommen: Unsinnige Aktionen, wie zum Beispiel das Entfernen der Diskette aus dem Laufwerk während des Kopiervorgangs, sind nicht berücksichtigt. Für diese und ähnliche Vorgänge ließen sich natürlich ebenfalls Sicherungen einbauen, doch der dafür betriebene Aufwand stünde in keiner Relation zum Nutzen.

Der Einfachheit halber sind Dmanager beim Aufruf zwei Parameter zu übergeben. Dabei bezieht sich der erste auf die verwendete Grafikkarte. Ohne Wissen um die Grafikkarte läßt sich der Inhalt des Ausgangsbildschirms nicht speichern. Eine korrekte Wiederherstellung dieses Bildschirms beim Abbruch des TSR-Dateimanagers wäre also nicht mehr gewährleistet.

Ebenso wie es gilt, das Programm vor möglichen Abstürzen zu bewahren, ist es auch wichtig, die Bildschirminformation und die Cursor-Position des ursprünglichen Programms zu sichern. Damit steht Ihnen nach Verlassen von Dmanager der ursprüngliche Bildschirm wieder zur Verfügung, und Sie können dort weiterarbeiten, wo Sie Ihre Anwendung zuvor unterbrochen haben.

Zu diesem Zweck ist die Bildschirminformation aus dem Bildschirmspeicher zu lesen und in einem Variablen-Array abzulegen. Da dieses Verfahren sehr schnell arbeitet, wird es auch angewendet, um die Bildschirmmaske des TSR-Dateimanagers nach den einzelnen Aktionen schnell wieder aufzubauen.

Beim Zwischenspeichern zweier Bildschirme ergeben sich dabei allerdings zwangsläufig zu viele Variablen. Deshalb speichert Dmanager den ursprünglichen Bildschirm in einer Datei auf einem Laufwerk. Das Laufwerk ist vorzugsweise eine RAM-Disk oder eine Festplatte, weil hier das Speichern und spätere Wiederabrufen der Bildschirminformation wesentlich schneller erfolgt als bei einem Diskettenlaufwerk. Eine RAM-Disk oder Festplatte ist jedoch nicht unbedingt notwendig, denn Dmanager sucht selbständig ein Laufwerk, auf dem die benötigten 4 KByte Speicherplatz zur Aufnahme der Bildschirmdatei »xxxxxxxx.xxx« vorhanden sind. Bei der Suche nach einem Laufwerk mit ausreichender Speicherkapazität wird mit dem Laufwerk F begonnen und in absteigender Reihenfolge gesucht. Spätestens bei Laufwerk A muß die Suche allerdings erfolgreich sein, weil die Bildschirmsicherung sich sonst nicht durchführen läßt. Bei Abschluß des Programms wird die Bildschirminformation in gleicher Weise auf den Laufwerken gesucht, gelesen und die Bildschirminformation in den Bildschirmspeicher zurückgeschrieben. Anschließend löscht Dmanager die nun nicht mehr benötigte Datei »xxxxxxxx.xxx«. Letztlich ist nur noch der Cursor auf die vorher gespeicherte Position zu setzen, und schon ist alles beim alten.

Da der Cursor während des Ablaufs des TSR-Dateimanagers an- und abgeschaltet wird, kann es vorkommen, daß sich die Cursor-Form verändert. Wünschen Sie eine andere Cursor-Form als die hier vorgegebene, so müssen Sie nur die für die CGA-Karte gekennzeichneten Werte in der Prozedur »CursorOn« verändern. Besitzer einer Hercules-Karte haben mit der gleichen Prozedur die entsprechenden Werte für die Hercules-Karte zu ändern.
Bereits die beiden Prozeduren »CursorOn« und »CursorOff« – in ein eigenes Programm eingebunden – stellen ein kleines Utility dar, um den Cursor beliebig ein- und auszuschalten beziehungsweise seine Form zu verändern. Den Nutzen der Routinen manifestieren all diejenigen Programme, die den Cursor abschalten oder seine Form verändern und am Ende den ursprünglichen Zustand nicht wiederherstellen. Daneben ist es ganz praktisch, den Cursor aus- und einschalten zu können; zum Beispiel dann, wenn Sie sich ein Menüsystem für Ihre Festplattenverwaltung per Stapeldateien zusammengestellt haben.
Wichtig ist auch die Klärung des physikalischen, das heißt des tatsächlichen Vorhandenseins eines Laufwerks B. Zwar läßt sich bei Systemen mit nur einem Diskettenlaufwerk dieses sowohl als Laufwerk A als auch als Laufwerk B ansprechen, doch erkennt der Datenträger-Interrupt 13hex dieses Laufwerk nicht als »wirkliches« Diskettenlaufwerk an. Ein leeres Laufwerk führt in diesem Fall unter Umständen zum Programmabsturz, während es sonst nach kurzer Überprüfung einfach ignoriert wird.
Die Laufwerksüberprüfung ist einer der beiden Fälle, bei denen für normale Programmfunktionen auf eine Interrupt-Programmierung zurückgegriffen werden muß. Um eine Spur des Laufwerks zu überprüfen, kommt der Interrupt 13hex zum Einsatz. Durch Übergabe des Wertes 4 im AH-Register wählen Sie die Funktion »Diskette verifizieren«. Die Werte in den Registern DH, DL, CH, CL und AL bestimmen Diskettenseite, Laufwerk, Spur, Sektor und Anzahl der zu prüfenden Sektoren. Den Fehlercode, der sich nach dem Interrupt-Aufruf im AH-Register befindet, nutzt Dmanager dazu, das Vorhandensein eines Diskettenlaufwerks zu bestimmen. Anschließend wird durch Übergabe des Wertes 0 im AH-Register das Laufwerk zurückgesetzt.

Vorhandensein oder Nichtvorhandensein – das ist hier die Frage

Der zweite Fall, der den Einsatz eines Interrupt-Aufrufs notwendig macht, ist der, wenn das aktuelle Laufwerk beim Aufruf des TSR-Dateimanagers bestimmt werden soll. Hierfür wird die Funktion 19hex des Interrupt 21hex verwendet. Als Resultat des Funktionsaufrufs erhalten Sie die Laufwerkskennung im AL-Register zurück. Dabei steht der Wert 0 für Laufwerk A, 1 für Laufwerk B und so weiter.
Wenn Sie die einzelnen Prozeduren des Programms »dmanager.pas« (Listing) einmal genauer betrachten, so erkennen Sie schnell, daß zur Simulation der MS-DOS-Funktionen meist nur wenige Programmzeilen notwendig sind. Die vielen »nebensächlichen Kleinigkeiten« nehmen deutlich mehr Platz in Anspruch. Dies ist beim Programmieren nicht anders als im täglichen Leben, und überall nennt man das, was dabei herauskommt, Komfort. Beim TSR-Dateimanager zeigt sich der Komfort in der Benutzeroberfläche, die für einen Großteil der Programmlänge verantwortlich ist. Wie schon bei der Absicherung gegen eine Fehlbedienung ist auch hier an eine Einsparung des Speicherbedarfs zu denken. Jedoch würde dadurch dem Zweck des Programms, das schließlich dem komfortablen Arbeiten mit dem Computer dienen soll, entgegengearbeitet. Ein vorrangiges Ziel hinsichtlich der Gestaltung des Bildschirms besteht darin, möglichst alle wichtigen Informationen auf einer Bildschirmseite darzustellen. Diese Forderung bestimmt den Aufbau der Bildschirmmaske: Die oberste und die unterste Zeile des Bildschirms sind für allgemeine Angaben reserviert (Angabe des aktuellen Laufwerks, Inhalt der Bildschirmfenster, Datum und Zeit). Dazwischen befinden sich drei Fenster, von denen das erste den Inhalt des aktuellen Verzeichnisses nennt, während das zweite die Belegung der Datenträger in den verschiedenen Laufwerken anzeigt beziehungsweise Auskunft über die Bereitschaft der Laufwerke gibt. Das dritte Fenster schließlich ist eine Art Multifunktionsfenster: Zum einen ersehen Sie daraus die erlaubten Optionen des Programms und zum anderen werden – mit zwei Ausnahmen – die Ein- und Ausgaben aller Funktionen über dieses Fenster abgewickelt. Die Funktion »Datei anschauen« sowie der Notizzettel bilden die Ausnahmen. Wie leicht einzusehen ist, benötigen Sie hierbei die gesamte Bildschirmgröße.

Es gilt, möglichst viele Informationen zu bieten, ohne die Übersicht zu beeinträchtigen

Den Grundstock der Bildschirmanzeige bildet die Prozedur »Maske«. Sie baut den Rahmen auf, der die einzelnen Anzeigen aufnehmen soll. Die einzige Besonderheit dieser Prozedur, wie auch der darauffolgenden Prozedur »Auswahl_zeigen«, ist die Verwendung der Insline-Prozedur. Durch das Einfügen einer neuen Zeile wird die aktuelle um eine Zeile nach unten gerückt. Das Verfahren erlaubt es, auch die Spalte 80 in Zeile 25 zu beschreiben, ohne daß der nachfolgende Zeilenvorschub die Bildschirmmaske zerstört. Direkte Schreiboperationen in den Bildschirmspeicher erübrigen sich dadurch.
Nach dem Aufbau der Maske zeigt das linke Fenster die Dateien des aktuellen Laufwerks an (Bild). Die Prozeduren »Inhalt« (sie liest den Inhalt des aktuellen Laufwerks) und »Inhalt_zeigen« kommen hier zum Einsatz. Vorher wird jedoch die Funktion »DiskTest« aufgerufen, um zu verhindern, daß das Programm »abstürzt«, falls sich in dem Laufwerk momentan keine Diskette befindet. Ist dies der Fall, wird die Variable »Inh« auf »false« gesetzt. Eine Anzeige des Inhalts unterbleibt damit.
Ist das Laufwerk bereit oder handelt es sich um ein Festplattenlaufwerk, so liest das Programm mit Hilfe der Prozeduren »FindFirst« und »FindNext« das Inhaltsverzeichnis. Beim Lesen werden die Dateinamen einem Array zugeordnet, das 224 Einträge aufnimmt. Dies entspricht der Anzahl an Einträgen, die bei einer 5 1/4-Zoll-Diskette mit 1,2 MByte Speicherkapazität erlaubt sind. Unabhängig von der tatsächlichen Länge wird der Dateiname auf acht und die Dateierweiterung auf drei Zeichen erweitert. Die Größe der Datei wird der Variablen »Eintrag« zugeordnet. Sofern es sich jedoch um ein Verzeichnis handelt, erhält die Variable den Eintrag »<DIR>«.
Die Anzeige des Inhaltsverzeichnisses auf dem Bildschirm übernimmt die Prozedur »Inhalt_zeigen«. Sie zeigt den Inhalt jeweils in »Seiten« zu 23 Einträgen an. Wenn Sie mehr Einträge sehen möchten, läßt sich das Verzeichnis mit Hilfe der Tasten [PgDn] und [PgUp] sowie der Cursor-Tasten »durchblättern«. Die Steuerung übernimmt hierbei die Prozedur »Cursor«. Diese Prozedur definiert die Seite und die Position eines Leuchtbalkens.
Der Leuchtbalken dient unter anderem zur Auswahl der Dateien für die verschiedenen Funktionen. Erzeugt wird der Balken – nachdem die 23 Einträge der aktuellen Seite geschrieben wurden – durch einfaches Überschreiben

des Namens an der definierten Position in inverser Darstellung. Jede Bewegung des Leuchtbalkens hat stets den gleichen Ablauf zur Folge: Ausgabe der Seite in normaler Darstellung und anschließend inverses Überschreiben der ausgewählten Zeile.
Im mittleren Fenster erscheinen die Angaben für die sechs möglichen Laufwerke. Die dafür notwendigen Schritte erledigen die Prozeduren »Speicher« und »Speicher_anzeigen«. Um die Anzahl der Dateien zu bestimmen, wird wieder auf »FindFirst« und »FindNext« zurückgegriffen. Den belegten und den freien Platz auf den Datenträgern bestimmen »DiskSize« und »DiskFree«. Treten dabei oder beim anfänglichen »DiskTest« Fehler auf, so führt dies zur Ausgabe der Fehlermeldung »Laufwerk n. bereit«.
Im rechten Fenster finden Sie eine Liste der Funktionen, die Sie per Tastendruck in Dmanager aktivieren können. Zunächst ist da einmal der Wechsel des Laufwerks durch Eingabe des Laufwerksbuchstabens. Beim Aufruf dieser Funktion wird gleichzeitig ein Verzeichniswechsel in das Hauptverzeichnis des gewählten Laufwerks durchgeführt. Deshalb ist es auch nötig, beim Verlassen des TSR-Dateimanagers »von Hand« wieder in das Ausgangsverzeichnis zu wechseln.
Ähnlich wie der Wechsel des Laufwerks funktioniert auch der Verzeichniswechsel. Plazieren Sie dazu den Leuchtbalken auf das gewünschte Verzeichnis und drücken Sie [Enter]. Um in das übergeordnete Verzeichnis zu gelangen, wählen Sie das Verzeichnis »..«, und wenn Ihre Wahl auf »/« fällt, haben Sie sich für das Hauptverzeichnis entschieden. Steht der Leuchtbalken beim Betätigen der Enter-Taste auf einer Datei, so hat dies keinerlei Auswirkungen.
Die folgenden drei Funktionen (das heißt Kopieren, Umbenennen und Löschen) befassen sich mit der Manipulation von Dateien. Die Prozedur »Kopie« liest und schreibt untypisierte Dateien mit Hilfe von »BlockRead« und »BlockWrite«. Dadurch lassen sich alle Arten von Dateien (sowohl reine Textdateien als auch Programmdateien) bearbeiten. Beim Kopieren werden Blöcke von jeweils 2048 Byte Länge (2 KByte) gelesen und dann in die neue Datei geschrieben. Dies geschieht unabhängig von der Art der Zeichen so lange, bis in der Quelldatei keine Daten mehr zur Verfügung stehen. Satzende- beziehungsweise Dateiendezeichen erfahren in untypisierten Dateien keine spezielle Behandlung.

Wollen Sie sich mit den MS-DOS-Kommandos begnügen, oder darf es ein bißchen mehr sein?

Noch einfacher zu handhaben ist das Umbenennen beziehungsweise Löschen von Dateien. Hier kommen einfach die Turbo-Pascal-Prozeduren »Rename« und »Erase« zur Anwendung. Wie inzwischen bekannt sein dürfte, macht die Prozedur »Rename« keinen Unterschied zwischen Dateien und Verzeichnissen. Es ist also durchaus möglich, auch Verzeichnisse umzubenennen – ein Komfort, den jeder bald zu schätzen weiß, der schon einmal Lust verspürte, einen nicht besonders aussagekräftigen Verzeichnisnamen zu ändern.
Das Löschen von Dateien läuft unter Verwendung von »Erase« ähnlich ab. Wenn Sie »auf Nummer Sicher« gehen möchten, können Sie noch eine Sicherheitsabfrage in die Prozedur »Löschen« einfügen. Ob dies notwendig ist oder nicht, möge jeder für sich selbst entscheiden.
Ebenso einfach wie das Umbenennen und Löschen von Dateien funktioniert das Anlegen und Löschen von Verzeichnissen. »MkDir« und »RmDir« erledigen die Arbeit.
Allerdings besteht ein grundlegender Unterschied zwischen den Prozeduren »Verzeichnis« und »Entfernen«: Während Sie Verzeichnisse und Unterverzeichnisse jederzeit in jedem Laufwerk anlegen können, unabhängig vom aktuellen Laufwerk und Verzeichnis, lassen sich Verzeichnisse nur im aktuellen Laufwerk und Unterverzeichnisse nur im aktuellen Verzeichnis entfernen.
Der Unterschied ergibt sich aus der Art, wie »MkDir« und »RmDir« die Verzeichnisnamen mitgeteilt bekommen. Während beim Anlegen eines Verzeichnisses die Eingabe des Namens über die Tastatur erfolgt und sich somit völlig frei gestalten läßt, treffen Sie die Wahl beim Entfernen mit Hilfe des Leuchtbalkens. Daraus ergibt sich zwangsläufig die genannte Einschränkung bei der Auswahl.
Für die Prozeduren »Anschauen« und »Notizzettel« verwendet Dmanager den Dateityp »Text«. Die Art des Schreibens und Lesens fällt jedoch bei beiden Prozeduren unterschiedlich aus. Die Prozedur »Anschauen« liest den Inhalt der ausgewählten Datei zeichenweise und stellt ihn ebenso auf dem Bildschirm dar. Durch Tastendruck läßt sich die Ausgabe jederzeit anhalten und mit [Esc] abbrechen. Wollen Sie dabei etwas »schwarz auf weiß« vorliegen haben, so fertigen Sie sich mit [PrtSc] einen Bildschirmausdruck (Hardcopy) an.

Eine Alternative zu Notizblock und Bleistift neben Ihrer PC-Tastatur

Im Gegensatz zu »Anschauen« werden die Eingaben der Prozedur »Notizzettel« zeilenweise erfaßt. Auf diese Weise lassen sich innerhalb der Zeile, in der sich der Cursor bei der Eingabe befindet, mit Hilfe der Backspace-Taste Zeichen löschen und somit Eingabefehler korrigieren. Nach 78 Zeichen springt der Cursor automatisch in die nächste Zeile. Durch Drücken von [Enter] wechseln Sie (unabhängig von der gegenwärtigen Spaltenposition) in eine neue Zeile.
Bei der Länge der zu erzeugenden Datei gibt es dagegen keine Einschränkung. Wer viel zu notieren hat, kann dies auf einem beliebig »langen« Notizzettel tun. Nur die Kapazität des Datenträgers begrenzt Ihren Drang zum Schreiben.
Die letzte Funktion des TSR-Dateimanagers stellt einen Taschenrechner zur Verfügung. Ebenso wie der Notizzettel läßt sich auch der Taschenrechner leistungsfähiger gestalten. Ob dies sinnvoll ist, müssen Sie selbst entscheiden und gegebenenfalls die Prozedur verändern. In der vorliegenden Fassung beherrscht der Taschenrechner lediglich die vier Grundrechenarten. Mit elf signifikanten Dezimalstellen nimmt er es aber mit jedem »richtigen« Taschenrechner auf.
Ohne explizite Eingabe nimmt die Routine automatisch den Wert 0 beziehungsweise den Wert 1 an, sofern es sich um eine Division handelt. Dies gilt jedoch nur für die erste Rechnung. Bei jeder weiteren Rechnung ohne Eingabe wird als erste Zahl das Ergebnis der vorangegangenen Rechnung und als zweite Zahl die vorhergehende Zahl angenommen. Kettenrechnungen lassen sich somit problemlos bewältigen. Bei einer Division wird die zweite Zahl stets daraufhin überprüft, ob sie den Wert 0 angenommen hat. Gegebenenfalls wird dieser Wert dann durch 1 ersetzt.
Soviel zum Leistungsumfang von Dmanager. Was bleibt, ist die speicherresidente Installation des TSR-Dateimanagers. Mit Hilfe der Prozedur »Keep« ab Turbo Pascal 4.0 ist dies relativ einfach zu bewerkstelligen. Bis auf wenige Details folgt die Installation dem in der Ausgabe 4 der DOS Extra erläuterten Beispiel, das die Nutzung des Tastatur-Interrupt 16hex zum Aufruf speicherresidenter Programme beschreibt.

Ganz gleich, ob Sie den TSR-Dateimanager unverändert übernehmen oder ihn an Ihre speziellen Bedürfnisse anpassen: In jedem Fall haben Sie mit dem Programm Dmanager einen nützlichen Helfer bei der Arbeit mit dem Computer – und vermutlich werden Sie ihn schon bald nicht mehr missen wollen. (Siegfried Stößlein/zi)

Name	dmanager.pas
Sprache	Turbo Pascal ab Version 4.Ø
Grafikkarte	jede
Besonderheiten	keine

```
[795]    1: { Programmname: DMANAGER.PAS }
[V94]    2: { (C)  Everts & Hagedorn     }
[ZB4]    3: {      DOS International     }
[T16]    4: { Autor: Siegfried Stößlein  }
[ØØØ]    5:
[NT4]    6: PROGRAM TSR_Datei_Manager;
[ØØØ]    7:
[LØ2]    8: {$M $4ØØØ, Ø, Ø}
[ØØØ]    9:
[X82]   1Ø: uses Crt, Dos;
[ØØØ]   11:
[RC5]   12: const HotKey1 = 28928;  { Alt + F1Ø }
[8P4]   13:       HotKey2 = 26368;  { Ctrl + F1Ø }
[ØØØ]   14:
[726]   15: type Screen = array [1..4ØØØ] of byte;
[ØØØ]   16:
[RNØ]   17: var
[563]   18:    AltSP, AltSS : word;
[A64]   19:    TurboSP, TurboSS : word;
[E96]   2Ø:    IntProg, AltInt16, NeuInt16 : pointer;
[ZB3]   21:    a, b, c, d, i : integer;
[9L4]   22:    AltX, AltY, CursorY : byte;
[AD6]   23:    BildschirmCGA : Screen absolute $B8ØØ:Ø;
[ZB1]   24:      { CGA }
[YD6]   25:    BildschirmHer : Screen absolute $BØØØ:Ø;
[822]   26:      { Hercules }
[114]   27:    BildschirmNeu : Screen;
[GU4]   28:    BildschirmName : string[15];
[9D3]   29:    DirInfo : SearchRec;
[856]   3Ø:    Eintrag : array[1..224] of string[19];
[SN5]   31:    Verz    : array[1..224] of string[7];
[N24]   32:    Auswahl, Auswahl1 : char;
[GP2]   33:    Seite : integer;
[2K3]   34:    CursorPos : integer;
[H53]   35:    Wahl, Weiter : char;
[BS5]   36:    Jahr, Monat, Tag, Wochentag : word;
[AN3]   37:    Stunde, Minute : word;
[HR4]   38:    Sekunde, Hundertstel : word;
[N65]   39:    Datnr : array[1..6] of integer;
[S65]   4Ø:    Bel : array[1..6] of string[1Ø];
[C35]   41:    Frei : array[1..6] of string[1Ø];
[B82]   42:    Inh : boolean;
[WC4]   43:    Fehler1, Fehler2 : integer;
[CM2]   44:    Laufwerk : char;
[5U3]   45:    Datei1, Datei2 : string;
[8W1]   46:    f1, f2 : file;
[H43]   47:    f : file of Screen;
[3K1]   48:    ft : Text;
[DH2]   49:    Zeichen : char;
[NC5]   5Ø:    ZeichenK : array [1..2Ø48] of char;
[CL4]   51:    gelesen, geschrieben : word;
[XE4]   52:    Dir, Dir1, Dir2 : string[17];
[KB3]   53:    Datenende : boolean;
[3E3]   54:    Eingabe : string[8Ø];
[9K4]   55:    Zahl1, Zahl2, Ergebnis : real;
[ZC3]   56:    ZahlA, ZahlB : string;
[292]   57:    Symbol : char;
[ØS2]   58:    Regs : Registers;
[XE3]   59:    DiskFehler : integer;
[ØK5]   6Ø:    GraphDriver, GraphMode : integer;
[EW4]   61:    Parameter1, Parameter2 : char;
[1K5]   62:    Parameter1a, Parameter2a : string;
[ØØØ]   63:
[926]   64: PROCEDURE INT16(Flags, cs, ip, ax, bx, cx,
[SL4]   65:     dx, si, di, ds, es, bp : word);
[8X1]   66:       interrupt;
[2M5]   67:   Function AlterInterrupt : word;
[K95]   68:   inline($31/$cØ/$9c/$FF/$1E/AltInt16);
[MUØ]   69: begin
[4H2]   7Ø:   if hi(ax) = Ø then
[P31]   71:   repeat
[1H3]   72:     ax := AlterInterrupt;
[Z75]   73:     if (ax = HotKey1) or (ax = HotKey2)
[ZN1]   74:     then begin
[L85]   75:       inline($8c/$16/AltSS/$89/$26/AltSP/
[553]   76:              $FA/$8E/$16/TurboSS/
[483]   77:              $8B/$26/TurboSP/$FB);
[BU3]   78:       GetIntVec($16,NeuInt16);
[MT3]   79:       SetIntVec($16,AltInt16);
[MS3]   8Ø:       inline($ff/$1E/IntProg);
[3V3]   81:       SetIntVec($16,NeuInt16);
[SN4]   82:       inline($FA/$8E/$16/AltSS/$8B/$26/
[RR1]   83:              AltSP/$FB);
[ZNØ]   84:     end;
[S93]   85:     until (ax <> HotKey1)
[2N2]   86:       or (ax <> HotKey2)
[YT5]   87:     else inline($a1/altint16+2/$87/$46/$1Ø/
[SF3]   88:                 $8B/$1E/altint16/$87/$5E/
[Y43]   89:                 $ØE/$5D/$Ø7/$1F/$5F/$5E/
[CP1]   9Ø:                 $5a/$59/$CB);
[ZNØ]   91: end;
[ØØØ]   92:
[215]   93: PROCEDURE Bildschirm_merken;
[MUØ]   94: begin
[ML3]   95:   if Parameter1 = 'C' then
[X35]   96:     BildschirmNeu := BildschirmCGA
[ZRØ]   97:   else
[XC5]   98:     BildschirmNeu := BildschirmHer;
[ZNØ]   99: end;
[ØØØ]  1ØØ:
[PC5]  1Ø1: PROCEDURE Bildschirm_erinnern;
[MUØ]  1Ø2: begin
[ML3]  1Ø3:   if Parameter1 = 'C' then
[E35]  1Ø4:     BildschirmCGA := BildschirmNeu
[ZRØ]  1Ø5:   else
[EC5]  1Ø6:     BildschirmHer := BildschirmNeu;
[ZNØ]  1Ø7: end;
[ØØØ]  1Ø8:
[GW4]  1Ø9: FUNCTION DiskTest : integer;
[MUØ]  11Ø: begin
[DR4]  111:   if Parameter2 = '1' then if a = 2
[LR1]  112:     then exit;
[RØ2]  113:   with Regs do
[MUØ]  114:   begin
[TZØ]  115:     AH := 4;
[X76]  116:       { Sektor auf Diskette verifizieren }
[LZØ]  117:     DH := 1;
[Z22]  118:       { Seite 1 }
[XE1]  119:     DL := a - 1;
[7U4]  12Ø:       { Laufwerk: Ø = A, 1 = B usw. }
[HZØ]  121:     CH := 1;
[UR1]  122:       { Spur 1 }
[XZØ]  123:     CL := 1;
[RY1]  124:       { Sektor 1 }
[TZØ]  125:     AL := 1;
[P76]  126:       { 1 Sektor soll verifiziert werden }
[P72]  127:     Intr(19,Regs);
[345]  128:       { BIOS-Diskette1nterrupt }
[TK2]  129:     DiskTest := AH;
[Y16]  13Ø:       {Ausg. der Fehlernr. im AH-Register}
[9ZØ]  131:     AH := Ø;
[4X6]  132:       { Zurücksetzen des Diskettensystems }
[P72]  133:     Intr(19,Regs);
[ZNØ]  134:   end;
[ZNØ]  135: end;
[ØØØ]  136:
[865]  137: PROCEDURE Bildschirm_sichern;
[MUØ]  138: begin
[X93]  139:   Bildschirm_merken;
[M83]  14Ø:   for a := 6 downto 1 do
[MUØ]  141:   begin
[DR4]  142:     if Parameter2 = '1' then if a = 2
[SL1]  143:       then a := 1;
[GE2]  144:     DiskFehler := Ø;
[SY3]  145:     DiskFehler := DiskTest;
[ØF3]  146:     if DiskFehler < 2 then
[MUØ]  147:     begin
[WV1]  148:       Fehler1 := Ø;
[CU2]  149:       BildschirmName :=
[AW4]  15Ø:         chr(a+64) + ':\xxxxxxxx.xxx';
[5A4]  151:       Assign(f, BildschirmName);
[674]  152:       {$I-} Rewrite(f); {$I+}
[L93]  153:       Fehler1 := IOResult;
[GT2]  154:       if Fehler1 = Ø then
[MUØ]  155:       begin
[ØW1]  156:         Fehler2 := Ø;
[B86]  157:         {$I-} Write(f, BildschirmNeu);{$I+}
[R93]  158:         Fehler2 := IOResult;
[LT2]  159:         if Fehler2 = Ø then
[MUØ]  16Ø:         begin
[KD1]  161:           Close(f);
[ASØ]  162:           exit
[ZNØ]  163:         end;
[BC4]  164:         if Fehler2 <> Ø then Erase(f);
[4MØ]  165:       end
[4MØ]  166:     end
[4MØ]  167:   end
[ZNØ]  168: end;
[ØØØ]  169:
[S15]  17Ø: PROCEDURE Bildschirm_zeigen;
[MUØ]  171: begin
[P31]  172:   repeat
[WV1]  173:     Fehler1 := Ø;
```

```
[ØW1]  174:      Fehler2 := Ø;
[5A4]  175:      Assign(f, BildschirmName);
[5K3]  176:      {$I-} Reset(f); {$I+}
[L93]  177:      Fehler1 := IOResult;
[KW2]  178:      if Fehler1 <> Ø then
[MUØ]  179:      begin
[3YØ]  18Ø:        a := Ø;
[P31]  181:        repeat
[MB1]  182:          a := a + 1;
[DR4]  183:          if Parameter2 = '1' then if a = 2
[ØM1]  184:            then a := 3;
[GE2]  185:          DiskFehler := Ø;
[SY3]  186:          DiskFehler := DiskTest;
[ØF3]  187:          if DiskFehler < 2 then
[MUØ]  188:          begin
[ØW1]  189:            Fehler2 := Ø;
[HV2]  19Ø:            Assign(f, chr(a+64)
[AB3]  191:              + ':\xxxxxxxx.xxx');
[5K3]  192:            {$I-} Reset(f); {$I+}
[W73]  193:            Fehler2 := IOResult
[ZNØ]  194:          end;
[N85]  195:          if DiskFehler > 1 then Fehler2 := 1
[9U3]  196:        until (a=6) or (Fehler2=Ø)
[ZNØ]  197:      end;
[NW2]  198:      if Fehler2 <> Ø then
[MUØ]  199:      begin
[GH2]  2ØØ:        Window(1,1,8Ø,25);
[DB1]  2Ø1:        ClrScr;
[Ø76]  2Ø2:        Writeln('Ich kann den gespeicherten '
,
[3F4]  2Ø3:          'Bildschirm nicht finden.');
[EL5]  2Ø4:        Writeln('Bitte wechseln Sie die ',
[8T3]  2Ø5:          'Diskette in Laufwerk ',
[6U3]  2Ø6:          BildschirmName[1], ',');
[Z95]  2Ø7:        Writeln('und drücken Sie eine ',
[6E1]  2Ø8:          'Taste!');
[XF1]  2Ø9:        Writeln;
[RN5]  21Ø:        Writeln('Abbruch mit [ESC] - Die ',
[325]  211:          'Bildschirminformation geht ',
[HV2]  212:          'dann verloren.');
[SX2]  213:        Weiter := ReadKey;
[SØ4]  214:        if Weiter = #27 then exit;
[ZNØ]  215:      end;
[TK4]  216:      if Fehler2 = Ø then Fehler1 := Ø
[3U2]  217:    until Fehler1 = Ø;
[HV3]  218:    Read(f, BildschirmNeu);
[KD1]  219:    Close(f);
[GC1]  22Ø:    Erase(f);
[FR3]  221:    Bildschirm_erinnern;
[ZNØ]  222: end;
[ØØØ]  223:
[1A5]  224: PROCEDURE Laufwerk_bestimmen;
[MUØ]  225: begin
[RØ2]  226:   With Regs do
[MUØ]  227:   begin
[K61]  228:     AH := $19;
[9C2]  229:     Intr($21,Regs);
[FG1]  23Ø:     a := AL + 1;
[AL3]  231:     Laufwerk := Chr(a + 64);
[ZNØ]  232:   end;
[ZNØ]  233: end;
[ØØØ]  234:
[AK3]  235: PROCEDURE CursorOff;
[MUØ]  236: begin
[GC4]  237:   inline($B4/$Ø1/$B9/Ø/15/$CD/$1Ø);
[ZNØ]  238: end;
[ØØØ]  239:
[VF3]  24Ø: PROCEDURE CursorOn;
[MUØ]  241: begin
[ML3]  242:   if Parameter1 = 'C' then
[2R2]  243:     inline($B4/$Ø1/$B9/
[DX4]  244:            7/   { Endzeile des Cursors }
[9F5]  245:            6/   { Anfangszeile des Cursors}
[MØ1]  246:            $CD/$1Ø)
[ZRØ]  247:   else
[RE4]  248:   inline($B4/$Ø1/$B9/13/12/$CD/$1Ø);
[ZNØ]  249: end;
[ØØØ]  25Ø:
[UY2]  251: PROCEDURE InvOff;
[MUØ]  252: begin
[VC3]  253:   if Parameter1='C' then
[MUØ]  254:   begin
[RR3]  255:     TextColor(LightCyan);
[WF3]  256:     TextBackground(Blue)
[4MØ]  257:   end
[WP1]  258:   else begin
[M52]  259:     TextColor(7);
[8V2]  26Ø:     TextBackground(Ø)
[4MØ]  261:   end
[ZNØ]  262: end;
[ØØØ]  263:
[TV2]  264: PROCEDURE InvOn;
[MUØ]  265: begin
[VC3]  266:   if Parameter1='C' then
[MUØ]  267:   begin
[3X2]  268:     TextColor(Black);
[2F3]  269:     TextBackground(Cyan)
[4MØ]  27Ø:   end
[WP1]  271:   else begin
[S42]  272:     TextColor(Ø);
[FV2]  273:     TextBackground(7)
[4MØ]  274:   end
[ZNØ]  275: end;
[ØØØ]  276:
[6U2]  277: PROCEDURE Maske;
[MUØ]  278: begin
[GH2]  279:   Window(1,1,8Ø,25);
[DB1]  28Ø:   InvOff;
[DB1]  281:   ClrScr;
[G31]  282:   InvOn;
[D36]  283:   Write('Aktuelles LW:        Laufwerke ');
[XC4]  284:   for i := 1 to 49 do Write(' ');
[3Ø2]  285:   GotoXY(1,24);
[V64]  286:   Write('Datum:                          ');
[SD2]  287:   Write('Zeit:');
[VC4]  288:   for i := 1 to 47 do Write(' ');
[DB1]  289:   InvOff;
[PV1]  29Ø:   GotoXY(1,3);
[ZE1]  291:   InsLine;
[MP2]  292:   for a := 2 to 24 do
[MUØ]  293:   begin
[G12]  294:     GotoXY(21,a);
[LP1]  295:     Write(#179);
[FF5]  296:      {einfacher senkrechter Balken }
[U12]  297:     GotoXY(45,a);
[SM1]  298:     Write(#179)
[ZNØ]  299:   end;
[ES2]  3ØØ:   Window(22,2,45,24);
[AG2]  3Ø1:   for i := 1 to 6 do
[MUØ]  3Ø2:   begin
[DX2]  3Ø3:     Writeln(chr(i+64),
[EL3]  3Ø4:           '> Inhalt:      Dateien');
[494]  3Ø5:     Writeln('Bel.:          Byte');
[ZW3]  3Ø6:     Write('Frei:          Byte');
[WN3]  3Ø7:     if i < 6 then Writeln(#1Ø)
[ZNØ]  3Ø8:   end;
[ZNØ]  3Ø9: end;
[ØØØ]  31Ø:
[CF4]  311: PROCEDURE Auswahl_zeigen;
[MUØ]  312: begin
[ØK2]  313:   Window(46,1,8Ø,1);
[G31]  314:   InvOn;
[TS3]  315:   Write('Auswahl              ');
[DB1]  316:   InvOff;
[LS2]  317:   Window(46,2,8Ø,25);
[DB1]  318:   ClrScr;
[4C3]  319:   Writeln('A,B,C,D,E,F ',
[PP3]  32Ø:     '- Laufwerk wechseln');
[XF1]  321:   Writeln;
[6L3]  322:   Writeln(' [Enter]    ',
[T44]  323:     '- Verzeichnis wechseln');
[XF1]  324:   Writeln;
[AH5]  325:   Writeln('    K       - Datei kopieren');
[XF1]  326:   Writeln;
[WX2]  327:   Writeln('    U       ',
[CE3]  328:     '- Datei umbenennen');
[XF1]  329:   Writeln;
[KR5]  33Ø:   Writeln('    L       - Datei löschen');
[XF1]  331:   Writeln;
[ZX2]  332:   Writeln('    V       ',
[RV3]  333:     '- Verzeichnis anlegen');
[XF1]  334:   Writeln;
[LX2]  335:   Writeln('    R       ',
[614]  336:     '- Verzeichnis löschen');
[XF1]  337:   Writeln;
[PX2]  338:   Writeln('    S       ',
[833]  339:     '- Datei anschauen');
[XF1]  34Ø:   Writeln;
[Z55]  341:   Writeln('    N       - Notizzettel');
[XF1]  342:   Writeln;
[KM5]  343:   Writeln('    T       - Taschenrechner');
[G31]  344:   InvOn;
[U53]  345:   Write('     Auswahl ',
[SU3]  346:     'durch Tastendruck     ');
[DB1]  347:   InvOff;
[VZ4]  348:   Writeln(' Achten Sie darauf, ',
[TM2]  349:     'die Arbeit im');
[YC5]  35Ø:   Writeln(' Ausgangsverzeichnis ',
[AF2]  351:     'zu beenden!');
[G31]  352:   InvOn;
[8U3]  353:   Write('                        Ende ',
[X12]  354:     'mit [ESC]');
[DB1]  355:   InvOff;
[LZ1]  356:   GotoXY(1,2Ø);
[ZE1]  357:   InsLine;
[ZNØ]  358: end;
[ØØØ]  359:
[2V2]  36Ø: PROCEDURE Datum;
[MUØ]  361: begin
[US2]  362:   Window(1,25,8Ø,25);
[G31]  363:   InvOn;
[U26]  364:   GetDate(Jahr, Monat, Tag, Wochentag);
[SV1]  365:   GotoXY(9,1);
[VT2]  366:   case Wochentag of
[HP2]  367:     Ø :  Write('Son');
[7P2]  368:     1 :  Write('Mon');
[1M2]  369:     2 :  Write('Die');
```

Listing. Der TSR-Dateimanager »dmanager.pas« bewältigt die üblichen Dateiverwaltungsfunktionen

```
[HP2]  37Ø:      3 :  Write('Mit');
[VN2]  371:      4 :  Write('Don');
[6N2]  372:      5 :  Write('Fre');
[FN2]  373:      6 :  Write('Sam');
[ZNØ]  374:    end;
[NZ1]  375:    GotoXY(14,1);
[185]  376:    Write(Tag:2,'.', Monat:2,'.',Jahr);
[141]  377:    InvOff
[ZNØ]  378: end;
[ØØØ]  379:
[2T2]  38Ø: PROCEDURE Zeit;
[MUØ]  381: begin
[8G5]  382:   GetTime(Stunde, Minute, Sekunde,
[MB2]  383:     Hundertstel);
[US2]  384:   Window(1,25,8Ø,25);
[G31]  385:   InvOn;
[ZZ1]  386:   GotoXY(38,1);
[2W4]  387:   Write(Stunde:2,':',Minute:2,':',
[RV1]  388:     Sekunde:2);
[DB1]  389:   InvOff;
[ZNØ]  39Ø: end;
[ØØØ]  391:
[WD3]  392: PROCEDURE Speicher;
[MUØ]  393: begin
[UF2]  394:   for a := 1 to 6 do
[MUØ]  395:   begin
[P31]  396:     repeat
[K82]  397:       Datnr[a] := Ø;
[DL3]  398:       Bel[a] := ' Laufwerk ';
[5S3]  399:       Frei[a] := ' n. bereit';
[DR4]  4ØØ:       if Parameter2 = '1' then if a = 2
[ØM1]  4Ø1:         then a := 3;
[GE2]  4Ø2:       DiskFehler := Ø;
[SY3]  4Ø3:       DiskFehler := DiskTest;
[ER3]  4Ø4:       if DiskFehler = 128 then
[MUØ]  4Ø5:       begin
[MB1]  4Ø6:         a := a + 1;
[SU2]  4Ø7:         if a > 6 then exit;
[ZNØ]  4Ø8:       end;
[SA3]  4Ø9:     until DiskFehler < 2;
[RL5]  41Ø:     FindFirst(chr(a+64)+':*.*',Directory,
[XH1]  411:       DirInfo);
[K82]  412:     Datnr[a] := Ø;
[H63]  413:     while DosError = Ø do
[MUØ]  414:     begin
[NY3]  415:       Datnr[a] := Datnr[a]+1;
[U33]  416:       FindNext(DirInfo);
[ZNØ]  417:     end;
[LU2]  418:     if DiskSize(a) = -1
[D94]  419:       then Bel[a] := ' Laufwerk '
[XU5]  42Ø:       else Str(DiskSize(a)-DiskFree(a):1Ø,
[PA1]  421:         Bel[a]);
[TS2]  422:     if DiskFree(a) = -1
[3F4]  423:       then Frei[a] := ' n. bereit'
[FF5]  424:       else Str(DiskFree(a):1Ø,Frei[a]);
[ZNØ]  425:   end;
[ZNØ]  426: end;
[ØØØ]  427:
[435]  428: PROCEDURE Speicher_anzeigen;
[MUØ]  429: begin
[ES2]  43Ø:   Window(22,2,45,24);
[UF2]  431:   for a := 1 to 6 do
[MUØ]  432:   begin
[PG2]  433:     GotoXY(12,a*4-3);
[W23]  434:     Write(Datnr[a]:4);
[KR5]  435:     if Bel[a] = ' Laufwerk ' then InvOn;
[HB2]  436:     GotoXY(6,a*4-2);
[SB2]  437:     Write(Bel[a]);
[EB2]  438:     GotoXY(6,a*4-1);
[VN2]  439:     Write(Frei[a]);
[141]  44Ø:     InvOff
[4MØ]  441:   end
[ZNØ]  442: end;
[ØØØ]  443:
[UX2]  444: PROCEDURE Inhalt;
[MUØ]  445: begin
[FF3]  446:   a := Ord(Laufwerk) - 64;
[DR4]  447:   if Parameter2 = '1' then if a = 2
[ØM1]  448:     then a := 3;
[GE2]  449:   DiskFehler := Ø;
[SY3]  45Ø:   DiskFehler := DiskTest;
[ER3]  451:   if DiskFehler = 128 then
[MUØ]  452:   begin
[UØ2]  453:     Inh := false;
[ØW9]  454:     exit;
[ZNØ]  455:   end;
[YN2]  456:   for i := 1 to 224 do
[Z14]  457:     Eintrag[i] := '                      ';
[YR5]  458:   FindFirst('*.*',Directory,DirInfo);
[LYØ]  459:   i := Ø;
[H63]  46Ø:   while DosError = Ø do
[MUØ]  461:   begin
[541]  462:     i := i+1;
[SK4]  463:     Eintrag[i] := DirInfo.Name;
[KW3]  464:     b := Pos('.',Eintrag[i]);
[2X4]  465:     if b <> Ø then for c := Ø to 8-b do
[D14]  466:       insert(' ',Eintrag[i],b);
[WX3]  467:     d := length(Eintrag[i]);
[6T4]  468:     if d < 12 then for c := Ø to 11-d do
[AC4]  469:       insert(' ',Eintrag[i],d+1);
[EP2]  47Ø:     with DirInfo do
[X64]  471:     if Attr and Directory <> Ø
[TV3]  472:       then Verz[i] := '<DIR>  '
[3T5]  473:     else Str(DirInfo.Size:7,Verz[i]);
[FE5]  474:     Eintrag[i] := Concat(Eintrag[i],
[1P1]  475:       Verz[i]);
[KS4]  476:     if Eintrag[i] = '        .    <DIR>  '
[2RØ]  477:       then
[ØK4]  478:       Eintrag[i] := '        \    <DIR>  ';
[U33]  479:     FindNext(DirInfo);
[ZNØ]  48Ø:   end;
[ZNØ]  481: end;
[ØØØ]  482:
[X94]  483: PROCEDURE Inhalt_zeigen;
[MUØ]  484: begin
[CH2]  485:   Window(1,2,2Ø,24);
[DB1]  486:   ClrScr;
[ØU3]  487:   if Inh = false then exit;
[GV1]  488:   GotoXY(1,1);
[EC5]  489:   for i := 23*Seite-22 to 23*Seite do
[MUØ]  49Ø:   begin
[Z53]  491:     Write(Eintrag[i]);
[DA4]  492:     if i mod 23 <> Ø then Writeln
[ZNØ]  493:   end;
[ZNØ]  494: end;
[ØØØ]  495:
[XZ2]  496: PROCEDURE Cursor;
[MUØ]  497: begin
[ØU3]  498:   if Inh = false then exit;
[CH2]  499:   Window(1,2,2Ø,24);
[EG3]  5ØØ:   GotoXY(1,CursorPos);
[9F6]  5Ø1:   Write(Eintrag[23*Seite-23+CursorPos]);
[LX4]  5Ø2:   if (Auswahl=#Ø) or (Wahl=#Ø) then
[MUØ]  5Ø3:   begin
[ØD3]  5Ø4:     Auswahl1 := Readkey;
[YD2]  5Ø5:     if Auswahl1 = #72
[C45]  5Ø6:       then CursorPos := CursorPos-1;
[VD2]  5Ø7:     if Auswahl1 = #8Ø
[A45]  5Ø8:       then CursorPos := CursorPos+1;
[ØE2]  5Ø9:     if Auswahl1 = #73
[2B5]  51Ø:       then CursorPos := CursorPos-23;
[XD2]  511:     if Auswahl1 = #81
[ØB5]  512:       then CursorPos := CursorPos+23;
[ZNØ]  513:   end;
[7B3]  514:   if CursorPos < 1 then
[MUØ]  515:   begin
[4L2]  516:     Seite := Seite-1;
[MY4]  517:     if Seite < 1 then CursorPos := 1
[NB3]  518:     else CursorPos := 23;
[S84]  519:     if Seite < 1 then Seite := 1;
[9D2]  52Ø:     Inhalt_zeigen
[ZNØ]  521:   end;
[3G3]  522:   if CursorPos > 23 then
[MUØ]  523:   begin
[4Y3]  524:     if (Eintrag[23*Seite+1] =
[2M2]  525:       '                      ') then
[MUØ]  526:     begin
[BK2]  527:       CursorPos := 23;
[4MØ]  528:     end
[ZRØ]  529:     else
[MUØ]  53Ø:     begin
[DH2]  531:       CursorPos := 1;
[6F2]  532:       Seite := Seite+1
[ZNØ]  533:     end;
[9D2]  534:     Inhalt_zeigen
[ZNØ]  535:   end;
[G31]  536:   InvOn;
[EG3]  537:   GotoXY(1,CursorPos);
[9F6]  538:   Write(Eintrag[23*Seite-23+CursorPos]);
[141]  539:   InvOff
[ZNØ]  54Ø: end;
[ØØØ]  541:
[ØØØ]  542:
[255]  543: PROCEDURE Laufwerk_anzeigen;
[MUØ]  544: begin
[GH2]  545:   Window(1,1,8Ø,25);
[TZ1]  546:   GotoXY(16,1);
[G31]  547:   InvOn;
[GS2]  548:   Write(Laufwerk);
[141]  549:   InvOff
[ZNØ]  55Ø: end;
[ØØØ]  551:
[845]  552: PROCEDURE Laufwerk_wechseln;
[MUØ]  553: begin
[MK3]  554:   Laufwerk_bestimmen;
[GE2]  555:   DiskFehler := Ø;
[SY3]  556:   DiskFehler := DiskTest;
[HE3]  557:   a := Ord(Auswahl) - 64;
[DR4]  558:   if Parameter2 = '1' then if a = 2
[LR1]  559:     then exit;
[GE2]  56Ø:   DiskFehler := Ø;
[SY3]  561:   DiskFehler := DiskTest;
[ZR4]  562:   if DiskFehler = 128 then exit;
[GF3]  563:   Laufwerk := Auswahl;
[DX1]  564:   Inh := true;
[WV1]  565:   Fehler1 := Ø;
[X93]  566:   Bildschirm_merken;
[8E5]  567:   {$I-} ChDir(Laufwerk+':\'); {$I+}
```

```
[FR3]  568:    Bildschirm_erinnern;
[L93]  569:    Fehler1 := IOResult;
[W15]  57Ø:    if Fehler1 <> Ø then Inh := false;
[VN1]  571:    Seite := 1;
[DH2]  572:    CursorPos := 1;
[ZNØ]  573: end;
[ØØØ]  574:
[P33]  575: PROCEDURE Anzeige;
[MUØ]  576: begin
[X93]  577:    Bildschirm_merken;
[DH1]  578:    Speicher;
[XA1]  579:    Inhalt;
[FR3]  58Ø:    Bildschirm_erinnern;
[B93]  581:    Laufwerk_anzeigen;
[X63]  582:    Speicher_anzeigen;
[UK2]  583:    Inhalt_zeigen;
[ØD1]  584:    Cursor;
[ZNØ]  585: end;
[ØØØ]  586:
[3P3]  587: PROCEDURE Ausfuehren;
[MUØ]  588: begin
[V35]  589:    Dir := Copy(Eintrag[23*Seite-23+
[ST2]  59Ø:      CursorPos],14,3);
[N75]  591:    Dir1 := Copy(Eintrag[23*Seite-23+
[LT2]  592:      CursorPos],1,12);
[P75]  593:    Dir2 := Copy(Eintrag[23*Seite-23+
[PN2]  594:      CursorPos],9,2);
[ZR2]  595:    if Dir = 'DIR' then
[MUØ]  596:    begin
[VN1]  597:      Seite := 1;
[DH2]  598:      CursorPos := 1;
[MN2]  599:      if Dir2= '\ ' then
[MUØ]  6ØØ:      begin
[ØM1]  6Ø1:        ChDir('\');
[FE1]  6Ø2:        Anzeige;
[ASØ]  6Ø3:        exit
[ZNØ]  6Ø4:      end;
[AG2]  6Ø5:      if Dir2= '..' then
[MUØ]  6Ø6:      begin
[DK1]  6Ø7:        ChDir('..');
[FE1]  6Ø8:        Anzeige;
[ASØ]  6Ø9:        exit
[ZNØ]  61Ø:      end;
[2U1]  611:      ChDir(Dir1);
[FE1]  612:      Anzeige;
[ZNØ]  613:    end;
[ZNØ]  614: end;
[ØØØ]  615:
[ØU2]  616: PROCEDURE Kopie;
[MUØ]  617: begin
[ØK2]  618:    Window(46,1,8Ø,1);
[G31]  619:    InvOn;
[UA4]  62Ø:    Write('Datei kopieren      ');
[DB1]  621:    InvOff;
[P31]  622:    repeat
[HS2]  623:      Window(46,2,8Ø,24);
[DB1]  624:      ClrScr;
[825]  625:      Writeln('Quelle mit Cursor');
[DX4]  626:      Writeln('auswählen und mit');
[YE5]  627:      Writeln('[Enter] bestätigen');
[X93]  628:      Bildschirm_merken;
[P31]  629:      repeat
[P31]  63Ø:        repeat
[VRØ]  631:          Zeit
[K13]  632:        until KeyPressed;
[YM2]  633:        Wahl := ReadKey;
[4N3]  634:        if Wahl = #Ø then Cursor
[9P4]  635:      until (Wahl=#13) or (Wahl=#27);
[RK2]  636:      if Wahl = #27 then
[MUØ]  637:      begin
[VU2]  638:        Auswahl_zeigen;
[ASØ]  639:        exit
[ZNØ]  64Ø:      end;
[ZM4]  641:      Datei1 := Eintrag[23*Seite-23+
[C72]  642:        CursorPos];
[VX2]  643:      Delete(Datei1,13,7);
[D75]  644:      for i := 1 to 8 do if Datei1[i] = ' '
[HØ4]  645:        then Delete(Datei1,i,9-i);
[HS2]  646:      Window(46,2,8Ø,24);
[P31]  647:      repeat
[5W1]  648:        GotoXY(1,8);
[F65]  649:        Writeln('Quelldatei: ', Datei1);
[WA3]  65Ø:        Write('Zieldatei : ');
[MN1]  651:        CursorOn;
[P31]  652:        repeat
[LØ2]  653:          GotoXY(13,9);
[V62]  654:          Readln(Datei2)
[YZ2]  655:        until Datei2 <> '';
[LV1]  656:        CursorOff;
[DT4]  657:        for i := 1 to Length(Datei2) do
[RD3]  658:        if Datei2[i] = ' ' then
[MUØ]  659:        begin
[WV1]  66Ø:          GotoXY(1,5);
[ØN4]  661:          Writeln('Falsche Eingabe!!!');
[FG4]  662:          Write('Zieldatei bitte neu ',
[YX1]  663:            'eingeben:');
[LØ2]  664:          GotoXY(13,9);
[DT4]  665:          for i := 1 to Length(Datei2) do
[3D1]  666:            Write(' ')
[ZNØ]  667:        end;
```

```
[BN5]  668:      until Datei2[Length(Datei2)] <> ' ';
[7Z4]  669:      if (Datei2[Length(Datei2)]=':')
[V25]  67Ø:        or (Datei2[Length(Datei2)]='\')
[YP4]  671:        then Datei2 := Datei2 + Datei1;
[GE2]  672:      DiskFehler := Ø;
[315]  673:      a := Ord(UpCase(Datei2[1])) - 64;
[TK5]  674:      if Parameter2 = '1' then if a = 2 then
[VH2]  675:        DiskFehler := 128
[ZRØ]  676:      else
[SY3]  677:      DiskFehler := DiskTest;
[ER3]  678:      if DiskFehler = 128 then
[MUØ]  679:      begin
[VU2]  68Ø:        Auswahl_zeigen;
[ASØ]  681:        exit
[ZNØ]  682:      end;
[MV2]  683:      Assign(f1,Datei1);
[WV1]  684:      Fehler1 := Ø;
[HP3]  685:      {$I-} Reset(f1,1); {$I+}
[L93]  686:      Fehler1 := IOResult;
[RV2]  687:      Assign(f2,Datei2);
[ØW1]  688:      Fehler2 := Ø;
[394]  689:      {$I-} Rewrite(f2,1); {$I+}
[R93]  69Ø:      Fehler2 := IOResult;
[WA5]  691:      if (Fehler1<>Ø) or (Fehler2<>Ø) then
[MUØ]  692:      begin
[FR3]  693:      Bildschirm_erinnern;
[HS2]  694:        Window(46,2,8Ø,24);
[SZ1]  695:        GotoXY(1,12);
[NL3]  696:        Write('*  F e h l e r  *');
[ZNØ]  697:      end;
[K45]  698:      if (Fehler1=Ø) and (Fehler2=Ø) then
[P31]  699:      repeat
[3V3]  7ØØ:        BlockRead(f1, ZeichenK,
[EH4]  7Ø1:          SizeOf(ZeichenK), gelesen);
[9B5]  7Ø2:        BlockWrite(f2, ZeichenK, gelesen,
[F42]  7Ø3:          geschrieben)
[6W2]  7Ø4:      until (gelesen = Ø)
[LE4]  7Ø5:        or (gelesen <> geschrieben);
[K45]  7Ø6:      if (Fehler1=Ø) and (Fehler2=Ø) then
[MUØ]  7Ø7:      begin
[7K1]  7Ø8:        Close(f1);
[RC1]  7Ø9:        Close(f2)
[ZNØ]  71Ø:      end;
[K45]  711:      if (Fehler1=Ø) and (Fehler2=Ø) then
[FR3]  712:      Bildschirm_erinnern;
[FE1]  713:      Anzeige;
[HS2]  714:      Window(46,2,8Ø,24);
[LZ1]  715:      GotoXY(1,2Ø);
[T74]  716:      Write('Noch einmal? (j/n)');
[V74]  717:      Weiter := UpCase(ReadKey);
[N23]  718:    until Weiter = 'N';
[FL2]  719:    Auswahl_zeigen
[ZNØ]  72Ø: end;
[ØØØ]  721:
[AP3]  722: PROCEDURE Umbenennen;
[MUØ]  723: begin
[ØK2]  724:    Window(46,1,8Ø,1);
[G31]  725:    InvOn;
[HL4]  726:    Write('Datei umbenennen    ');
[DB1]  727:    InvOff;
[P31]  728:    repeat
[HS2]  729:      Window(46,2,8Ø,24);
[DB1]  73Ø:      ClrScr;
[KG4]  731:      Writeln('Datei mit Cursor');
[DX4]  732:      Writeln('auswählen und mit');
[YE5]  733:      Writeln('[Enter] bestätigen');
[X93]  734:      Bildschirm_merken;
[P31]  735:      repeat
[P31]  736:        repeat
[VRØ]  737:          Zeit
[K13]  738:        until KeyPressed;
[YM2]  739:        Wahl := ReadKey;
[4N3]  74Ø:        if Wahl = #Ø then Cursor
[9P4]  741:      until (Wahl=#13) or (Wahl=#27);
[RK2]  742:      if Wahl = #27 then
[MUØ]  743:      begin
[VU2]  744:        Auswahl_zeigen;
[ASØ]  745:        exit
[ZNØ]  746:      end;
[ZM4]  747:      Datei1 := Eintrag[23*Seite-23+
[C72]  748:        CursorPos];
[VX2]  749:      Delete(Datei1,13,7);
[D75]  75Ø:      for i := 1 to 8 do if Datei1[i] = ' '
[HØ4]  751:        then Delete(Datei1,i,9-i);
[HS2]  752:      Window(46,2,8Ø,24);
[P31]  753:      repeat
[ØW1]  754:        Fehler2 := Ø;
[5W1]  755:        GotoXY(1,8);
[DZ4]  756:        Writeln('Alter Name: ', Datei1);
[MA3]  757:        Write('Neuer Name: ');
[MN1]  758:        CursorOn;
[P31]  759:        repeat
[LØ2]  76Ø:          GotoXY(13,9);
[V62]  761:          Readln(Datei2)
[YZ2]  762:        until Datei2 <> '';
[LV1]  763:        CursorOff;
[DT4]  764:        for i := 1 to Length(Datei2) do
```

Listing. Der TSR-Dateimanager »dmanager.pas« bewältigt die üblichen Dateiverwaltungsfunktionen

```
[TP2]  765:        if (Datei2[i]=':')
[BT2]  766:          or (Datei2[i]='\')
[UK3]  767:          or (Datei2[i]=' ') then
[MUØ]  768:        begin
[WV1]  769:          GotoXY(1,5);
[1W1]  77Ø:          Fehler2 := 1;
[ØN4]  771:          Writeln('Falsche Eingabe!!!');
[ZX4]  772:          Write('Neuen Namen bitte neu ',
[YX1]  773:            'eingeben:');
[LØ2]  774:          GotoXY(13,9);
[DT4]  775:          for i := 1 to Length(Datei2) do
[3D1]  776:            Write(' ')
[ZNØ]  777:        end;
[DØ3]  778:      until Fehler2 <> 1;
[MV2]  779:      Assign(f1,Datei1);
[WV1]  78Ø:      Fehler1 := Ø;
[2Y4]  781:      {$I-} Rename(f1,Datei2); {$I+}
[L93]  782:      Fehler1 := IOResult;
[KW2]  783:      if Fehler1 <> Ø then
[MUØ]  784:      begin
[FR3]  785:      Bildschirm_erinnern;
[HS2]  786:        Window(46,2,8Ø,24);
[SZ1]  787:        GotoXY(1,12);
[NL3]  788:        Write('*  F e h l e r  *');
[ZNØ]  789:      end;
[FE1]  79Ø:      Anzeige;
[HS2]  791:      Window(46,2,8Ø,24);
[LZ1]  792:      GotoXY(1,2Ø);
[T74]  793:      Write('Noch einmal? (j/n)');
[V74]  794:      Weiter := UpCase(ReadKey);
[N23]  795:    until Weiter = 'N';
[FL2]  796:    Auswahl_zeigen
[ZNØ]  797: end;
[ØØØ]  798:
[HD3]  799: PROCEDURE Loeschen;
[MUØ]  8ØØ: begin
[ØK2]  8Ø1:   Window(46,1,8Ø,1);
[G31]  8Ø2:   InvOn;
[YF4]  8Ø3:   Write('Datei löschen       ');
[DB1]  8Ø4:   InvOff;
[P31]  8Ø5:   repeat
[HS2]  8Ø6:     Window(46,2,8Ø,24);
[DB1]  8Ø7:     ClrScr;
[KG4]  8Ø8:     Writeln('Datei mit Cursor');
[DX4]  8Ø9:     Writeln('auswählen und mit');
[YE5]  81Ø:     Writeln('[Enter] bestätigen');
[X93]  811:     Bildschirm_merken;
[P31]  812:     repeat
[P31]  813:       repeat
[VRØ]  814:         Zeit
[K13]  815:       until KeyPressed;
[YM2]  816:       Wahl := ReadKey;
[4N3]  817:       if Wahl = #Ø then Cursor
[9P4]  818:     until (Wahl=#13) or (Wahl=#27);
[RK2]  819:     if Wahl = #27 then
[MUØ]  82Ø:     begin
[VU2]  821:       Auswahl_zeigen;
[ASØ]  822:       exit
[ZNØ]  823:     end;
[ZM4]  824:     Datei1 := Eintrag[23*Seite-23+
[C72]  825:       CursorPos];
[VX2]  826:     Delete(Datei1,13,7);
[D75]  827:     for i := 1 to 8 do if Datei1[i] = ' '
[HØ4]  828:       then Delete(Datei1,i,9-i);
[HS2]  829:     Window(46,2,8Ø,24);
[5W1]  83Ø:     GotoXY(1,8);
[384]  831:     Writeln('Datei: ', Datei1);
[MV2]  832:     Assign(f1,Datei1);
[WV1]  833:     Fehler1 := Ø;
[ZL3]  834:     {$I-} Erase(f1); {$I+}
[L93]  835:     Fehler1 := IOResult;
[KW2]  836:     if Fehler1 <> Ø then
[MUØ]  837:     begin
[FR3]  838:     Bildschirm_erinnern;
[HS2]  839:       Window(46,2,8Ø,24);
[SZ1]  84Ø:       GotoXY(1,12);
[NL3]  841:       Write('*  F e h l e r  *');
[ZNØ]  842:     end;
[FE1]  843:     Anzeige;
[HS2]  844:     Window(46,2,8Ø,24);
[LZ1]  845:     GotoXY(1,2Ø);
[T74]  846:     Write('Noch einmal? (j/n)');
[V74]  847:     Weiter := UpCase(ReadKey);
[N23]  848:   until Weiter = 'N';
[FL2]  849:   Auswahl_zeigen
[ZNØ]  85Ø: end;
[ØØØ]  851:
[XY3]  852: PROCEDURE Verzeichnis;
[MUØ]  853: begin
[ØK2]  854:   Window(46,1,8Ø,1);
[G31]  855:   InvOn;
[S74]  856:   Write('Verz. anlegen       ');
[DB1]  857:   InvOff;
[KP1]  858:   Datei1 := '';
[HS2]  859:   Window(46,2,8Ø,24);
[P31]  86Ø:   repeat
[DB1]  861:     ClrScr;
[3Y4]  862:     Writeln('Geben Sie bitte den ',
[G22]  863:       'Namen ein:');
[KH6]  864:     Writeln('(Laufwerk:Verzeichnisname)');
```

```
[XF1]  865:     Writeln;
[EU3]  866:     Write('Verzeichnis: ');
[P31]  867:     repeat
[VRØ]  868:       Zeit
[K13]  869:     until KeyPressed;
[HS2]  87Ø:     Window(46,2,8Ø,24);
[1Ø2]  871:     GotoXY(14,4);
[MN1]  872:     CursorOn;
[5E2]  873:     Readln(Datei1);
[LV1]  874:     CursorOff;
[GE2]  875:     DiskFehler := Ø;
[Ø15]  876:     a := Ord(UpCase(Datei1[1])) - 64;
[TK5]  877:     if Parameter2 = '1' then if a = 2 then
[VH2]  878:       DiskFehler := 128
[ZRØ]  879:     else
[SY3]  88Ø:     DiskFehler := DiskTest;
[ER3]  881:     if DiskFehler = 128 then
[MUØ]  882:     begin
[VU2]  883:       Auswahl_zeigen;
[ASØ]  884:       exit
[ZNØ]  885:     end;
[X93]  886:     Bildschirm_merken;
[WV1]  887:     Fehler1 := Ø;
[UC4]  888:     {$I-} MkDir(Datei1); {$I+}
[L93]  889:     Fehler1 := IOResult;
[FR3]  89Ø:     Bildschirm_erinnern;
[KW2]  891:     if Fehler1 <> Ø then
[MUØ]  892:     begin
[HS2]  893:       Window(46,2,8Ø,24);
[SZ1]  894:       GotoXY(1,12);
[NL3]  895:       Write('*  F e h l e r  *');
[ZNØ]  896:     end;
[FE1]  897:     Anzeige;
[HS2]  898:     Window(46,2,8Ø,24);
[LZ1]  899:     GotoXY(1,2Ø);
[T74]  9ØØ:     Write('Noch einmal? (j/n)');
[V74]  9Ø1:     Weiter := UpCase(ReadKey);
[N23]  9Ø2:   until Weiter = 'N';
[FL2]  9Ø3:   Auswahl_zeigen
[ZNØ]  9Ø4: end;
[ØØØ]  9Ø5:
[VG3]  9Ø6: PROCEDURE Entfernen;
[MUØ]  9Ø7: begin
[ØK2]  9Ø8:   Window(46,1,8Ø,1);
[G31]  9Ø9:   InvOn;
[GG4]  91Ø:   Write('Verz. entfernen     ');
[DB1]  911:   InvOff;
[KP1]  912:   Datei1 := '';
[P31]  913:   repeat
[HS2]  914:     Window(46,2,8Ø,24);
[DB1]  915:     ClrScr;
[EY5]  916:     Writeln('Verzeichnis mit Cursor');
[DX4]  917:     Writeln('auswählen und mit');
[YE5]  918:     Writeln('[Enter] bestätigen');
[X93]  919:     Bildschirm_merken;
[P31]  92Ø:     repeat
[P31]  921:       repeat
[VRØ]  922:         Zeit
[K13]  923:       until KeyPressed;
[YM2]  924:       Wahl := ReadKey;
[4N3]  925:       if Wahl = #Ø then Cursor
[9P4]  926:     until (Wahl=#13) or (Wahl=#27);
[RK2]  927:     if Wahl = #27 then
[MUØ]  928:     begin
[VU2]  929:       Auswahl_zeigen;
[ASØ]  93Ø:       exit
[ZNØ]  931:     end;
[V35]  932:     Dir := Copy(Eintrag[23*Seite-23+
[ST2]  933:       CursorPos],14,3);
[N75]  934:     Dir1 := Copy(Eintrag[23*Seite-23+
[LT2]  935:       CursorPos],1,12);
[ZR2]  936:     if Dir = 'DIR' then
[MUØ]  937:     begin
[N82]  938:       Datei1 := Dir1;
[VX2]  939:       Delete(Datei1,13,7);
[WV1]  94Ø:       Fehler1 := Ø;
[GD4]  941:       {$I-} RmDir(Datei1); {$I+}
[L93]  942:       Fehler1 := IOResult;
[ZNØ]  943:     end;
[FA5]  944:     if (Fehler1<>Ø) or (Dir<>'DIR') then
[MUØ]  945:     begin
[FR3]  946:     Bildschirm_erinnern;
[HS2]  947:       Window(46,2,8Ø,24);
[SZ1]  948:       GotoXY(1,12);
[NL3]  949:       Write('*  F e h l e r  *');
[ZNØ]  95Ø:     end;
[FE1]  951:     Anzeige;
[HS2]  952:     Window(46,2,8Ø,24);
[LZ1]  953:     GotoXY(1,2Ø);
[T74]  954:     Write('Noch einmal? (j/n)');
[V74]  955:     Weiter := UpCase(ReadKey);
[N23]  956:   until Weiter = 'N';
[FL2]  957:   Auswahl_zeigen
[ZNØ]  958: end;
[ØØØ]  959:
[9G3]  96Ø: PROCEDURE Anschauen;
[MUØ]  961: begin
[ØK2]  962:   Window(46,1,8Ø,1);
[G31]  963:   InvOn;
[WB4]  964:   Write('Datei anschauen     ');
```

```
[DB1]  965:    InvOff;
[P31]  966:    repeat
[HS2]  967:      Window(46,2,8Ø,24);
[DB1]  968:      ClrScr;
[KG4]  969:      Writeln('Datei mit Cursor');
[DX4]  97Ø:      Writeln('auswählen und mit');
[YE5]  971:      Writeln('[Enter] bestätigen');
[XF1]  972:      Writeln;
[ØU6]  973:      Writeln('Unterbrechen und Fortsetzen');
[ØK5]  974:      Writeln('der Bildschirmausgabe');
[LW4]  975:      Writeln('durch Tastendruck');
[XF1]  976:      Writeln;
[BV4]  977:      Writeln('Abbruch mit [ESC]');
[X93]  978:      Bildschirm_merken;
[P31]  979:      repeat
[P31]  98Ø:        repeat
[VRØ]  981:          Zeit
[K13]  982:        until KeyPressed;
[YM2]  983:        Wahl := ReadKey;
[4N3]  984:        if Wahl = #Ø then Cursor
[9P4]  985:      until (Wahl=#13) or (Wahl=#27);
[RK2]  986:      if Wahl = #27 then
[MUØ]  987:      begin
[VU2]  988:        Auswahl_zeigen;
[ASØ]  989:        exit
[ZNØ]  99Ø:      end;
[ZM4]  991:      Datei1 := Eintrag[23*Seite-23+
[C72]  992:        CursorPos];
[VX2]  993:      Delete(Datei1,13,7);
[D75]  994:      for i := 1 to 8 do if Datei1[i] = ' '
[HØ4]  995:        then Delete(Datei1,i,9-i);
[HS2]  996:      Window(46,2,8Ø,24);
[MZ1]  997:      GotoXY(1,11);
[384]  998:      Writeln('Datei: ', Datei1);
[UX2]  999:      Assign(ft,Datei1);
[WV1] 1ØØØ:      Fehler1 := Ø;
[5U3] 1ØØ1:      {$I-} Reset(ft); {$I+}
[L93] 1ØØ2:      Fehler1 := IOResult;
[KW2] 1ØØ3:      if Fehler1 <> Ø then
[MUØ] 1ØØ4:      begin
[FR3] 1ØØ5:        Bildschirm_erinnern;
[HS2] 1ØØ6:        Window(46,2,8Ø,24);
[ØØ2] 1ØØ7:        GotoXY(1,14);
[NL3] 1ØØ8:        Write('*  F e h l e r  *');
[4MØ] 1ØØ9:      end
[ZRØ] 1Ø1Ø:      else
[MUØ] 1Ø11:      begin
[GH2] 1Ø12:        Window(1,1,8Ø,25);
[DB1] 1Ø13:        ClrScr;
[T43] 1Ø14:        Datenende := false;
[P31] 1Ø15:        repeat
[7Y2] 1Ø16:          Read(ft, Zeichen);
[Y23] 1Ø17:          if KeyPressed then
[MUØ] 1Ø18:          begin
[SX2] 1Ø19:            Weiter := ReadKey;
[582] 1Ø2Ø:            if Weiter = #27
[ES3] 1Ø21:              then Datenende := true;
[543] 1Ø22:            if Datenende = false
[9W3] 1Ø23:              then Weiter := ReadKey;
[ZNØ] 1Ø24:          end;
[EM2] 1Ø25:          Write(Zeichen);
[LB5] 1Ø26:        until (Datenende=true) or Eof(ft);
[AL1] 1Ø27:        Close(ft);
[XF1] 1Ø28:        Writeln;
[XF1] 1Ø29:        Writeln;
[TY5] 1Ø3Ø:        Writeln('Drücken Sie eine Taste!');
[SX2] 1Ø31:        Weiter := ReadKey;
[FR3] 1Ø32:        Bildschirm_erinnern;
[GH2] 1Ø33:        Window(1,1,8Ø,25);
[H42] 1Ø34:        GotoXY(8Ø,25);
[ZNØ] 1Ø35:      end;
[FE1] 1Ø36:      Anzeige;
[HS2] 1Ø37:      Window(46,2,8Ø,24);
[LZ1] 1Ø38:      GotoXY(1,2Ø);
[T74] 1Ø39:      Write('Noch einmal? (j/n)');
[V74] 1Ø4Ø:      Weiter := UpCase(ReadKey);
[N23] 1Ø41:    until Weiter = 'N';
[FL2] 1Ø42:    Auswahl_zeigen
[ZNØ] 1Ø43: end;
[ØØØ] 1Ø44:
[B14] 1Ø45: PROCEDURE Notizzettel;
[MUØ] 1Ø46: begin
[X93] 1Ø47:   Bildschirm_merken;
[P31] 1Ø48:   repeat
[GH2] 1Ø49:     Window(1,1,8Ø,25);
[ØW1] 1Ø5Ø:     Fehler2 := Ø;
[P31] 1Ø51:     repeat
[P31] 1Ø52:       repeat
[XV1] 1Ø53:         Fehler1 := 1;
[DB1] 1Ø54:         ClrScr;
[KV3] 1Ø55:         Write('Notizzettel: ');
[MN1] 1Ø56:         CursorOn;
[NZ1] 1Ø57:         GotoXY(14,1);
[5E2] 1Ø58:         Readln(Datei1);
[8N4] 1Ø59:         if Datei1 = '' then Fehler2 := 1;
[9T4] 1Ø6Ø:         for i := 1 to Length(Datei1) do
[AK2] 1Ø61:           if Datei1[i] = ' '
[3U2] 1Ø62:             then Fehler1 := Ø;
[3Z4] 1Ø63:         if (Datei1[Length(Datei1)]=':')
[R25] 1Ø64:           or (Datei1[Length(Datei1)]='\')
[3U2] 1Ø65:           then Fehler1 := Ø;
[GT2] 1Ø66:       if Fehler1 = Ø then
[MUØ] 1Ø67:       begin
[WV1] 1Ø68:         GotoXY(1,5);
[3F4] 1Ø69:         Writeln('Falsche Eingabe!');
[PB5] 1Ø7Ø:         Write('Notizzetteldatei bitte ',
[6G2] 1Ø71:           'neu eingeben!');
[LV1] 1Ø72:         CursorOff;
[Y44] 1Ø73:         repeat until KeyPressed;
[MN1] 1Ø74:         CursorOn;
[ZNØ] 1Ø75:       end;
[5U2] 1Ø76:     until Fehler1 = 1;
[LT2] 1Ø77:     if Fehler2 = Ø then
[MUØ] 1Ø78:     begin
[GE2] 1Ø79:       DiskFehler := Ø;
[Ø15] 1Ø8Ø:       a := Ord(UpCase(Datei1[1])) - 64;
[DR4] 1Ø81:       if Parameter2 = '1' then if a = 2
[ØF3] 1Ø82:         then DiskFehler := 128
[ZRØ] 1Ø83:       else
[SY3] 1Ø84:       DiskFehler := DiskTest;
[ER3] 1Ø85:       if DiskFehler = 128 then
[MUØ] 1Ø86:       begin
[LV1] 1Ø87:         CursorOff;
[FR3] 1Ø88:       Bildschirm_erinnern;
[H42] 1Ø89:         GotoXY(8Ø,25);
[FE1] 1Ø9Ø:         Anzeige;
[VU2] 1Ø91:         Auswahl_zeigen;
[ASØ] 1Ø92:         exit
[ZNØ] 1Ø93:       end;
[2Ø3] 1Ø94:       Assign(ft, Datei1);
[WV1] 1Ø95:       Fehler1 := Ø;
[5U3] 1Ø96:       {$I-} Reset(ft); {$I+}
[L93] 1Ø97:       Fehler1 := IOResult;
[GT2] 1Ø98:       if Fehler1 = Ø then
[MUØ] 1Ø99:       begin
[WV1] 11ØØ:         GotoXY(1,5);
[Y25] 11Ø1:         Write('Datei existiert schon!',
[2E4] 11Ø2:           '  -  Überschreiben? (j/n)');
[V74] 11Ø3:         Weiter := UpCase(ReadKey);
[9Y4] 11Ø4:         if Weiter = 'J' then Fehler1 := 1
[ZNØ] 11Ø5:       end;
[ZNØ] 11Ø6:     end;
[8Ø3] 11Ø7:   until Fehler1 <> Ø;
[LT2] 11Ø8:   if Fehler2 = Ø then
[MUØ] 11Ø9:   begin
[WV1] 111Ø:     Fehler1 := Ø;
[EV3] 1111:     {$I-} Rewrite(ft); {$I+}
[L93] 1112:     Fehler1 := IOResult;
[KW2] 1113:     if Fehler1 <> Ø then
[MUØ] 1114:     begin
[ØØ2] 1115:       GotoXY(1,14);
[NL3] 1116:       Write('*  F e h l e r  *');
[4MØ] 1117:     end
[ZRØ] 1118:     else
[MUØ] 1119:     begin
[LV1] 112Ø:     GotoXY(1,2);
[FE4] 1121:     for i := 1 to 8Ø do Write(#2Ø5);
[NZ1] 1122:     GotoXY(5Ø,2);
[FR5] 1123:     Write(' Datei sichern mit [ESC] ');
[PH2] 1124:     Window(1,3,8Ø,25);
[DB1] 1125:     ClrScr;
[P31] 1126:       repeat
[LYØ] 1127:         i := Ø;
[H12] 1128:         Eingabe := '';
[G42] 1129:         Zeichen := #8;
[P31] 113Ø:         repeat
[WA2] 1131:           if (Zeichen=#8)
[VE2] 1132:             or (Zeichen=#27)
[483] 1133:             or (Zeichen>#31) then
[MUØ] 1134:           begin
[943] 1135:             CursorY := WhereY;
[H52] 1136:             if Zeichen = #8
[3F2] 1137:               then Eingabe :=
[PE2] 1138:                 Copy(Eingabe,1,
[5T2] 1139:                   Length(Eingabe)-1)
[PY3] 114Ø:             else Eingabe := Eingabe +
[WD1] 1141:               Zeichen;
[453] 1142:             GotoXY(1,CursorY);
[HY2] 1143:             Write(Eingabe + ' ');
[VZ3] 1144:             GotoXY(Length(Eingabe)+1,
[1K1] 1145:               CursorY)
[ZNØ] 1146:           end;
[763] 1147:           Zeichen := Readkey;
[F62] 1148:           if Zeichen > #31
[A42] 1149:             then i := i + 1;
[H52] 115Ø:           if Zeichen = #8
[E42] 1151:             then i := i - 1;
[DM2] 1152:           if i < Ø then i := Ø
[2Ø3] 1153:         until (Zeichen=#13)
[Z44] 1154:           or (Zeichen=#27) or (i=79);
[F62] 1155:         if Zeichen > #31
[TX3] 1156:           then Eingabe := Eingabe +
[WD1] 1157:             Zeichen;
[1H3] 1158:         Writeln(ft, Eingabe);
[GV3] 1159:         if (Zeichen=#13) or (i=79)
[392] 116Ø:           then Writeln;
[N43] 1161:       until Zeichen = #27;
```

Listing. Der TSR-Dateimanager »dmanager.pas« bewältigt die üblichen Dateiverwaltungsfunktionen

```
[TD1] 1162:          Close(ft)
[4MØ] 1163:        end
[ZNØ] 1164:      end;
[LV1] 1165:      CursorOff;
[XF1] 1166:      Writeln;
[XF1] 1167:      Writeln;
[T74] 1168:      Write('Noch einmal? (j/n)');
[V74] 1169:      Weiter := UpCase(ReadKey);
[N23] 117Ø:    until Weiter = 'N';
[FR3] 1171:    Bildschirm_erinnern;
[H42] 1172:    GotoXY(8Ø,25);
[FE1] 1173:    Anzeige;
[FL2] 1174:    Auswahl_zeigen
[ZNØ] 1175: end;
[ØØØ] 1176:
[ZC4] 1177: PROCEDURE Taschenrechner;
[MUØ] 1178: begin
[ØK2] 1179:   Window(46,1,8Ø,1);
[G31] 118Ø:   InvOn;
[7D4] 1181:   Write('Taschenrechner      ');
[DB1] 1182:   InvOff;
[HS2] 1183:   Window(46,2,8Ø,24);
[DB1] 1184:   ClrScr;
[9N5] 1185:   Writeln('Dieser Taschenrechner');
[4F4] 1186:   Writeln('beherrscht die');
[DY4] 1187:   Writeln('Grundrechenarten.');
[A26] 1188:   Writeln('Die Symbole zum Rechnen');
[CU3] 1189:   Writeln('sind:  +  -  *  /');
[6M1] 119Ø:   Zahl1 := Ø;
[8M1] 1191:   Zahl2 := Ø;
[T62] 1192:   Ergebnis := Ø;
[P31] 1193:   repeat
[BT2] 1194:     Window(46,1Ø,8Ø,24);
[DB1] 1195:     ClrScr;
[MN1] 1196:     CursorOn;
[7L2] 1197:     Write('Zahl1 : ');
[582] 1198:     Readln(ZahlA);
[YD5] 1199:     if ZahlA = '' then Zahl1 := Ergebnis
[VZ3] 12ØØ:       else Val(ZahlA, Zahl1, i);
[SV1] 12Ø1:     GotoXY(9,1);
[F53] 12Ø2:     Writeln(Zahl1:1:11);
[DR2] 12Ø3:     Write('Symbol: ');
[P31] 12Ø4:     repeat
[VT2] 12Ø5:       Symbol := ReadKey
[YV4] 12Ø6:     until (Symbol='+') or (Symbol='-')
[ZT4] 12Ø7:       or (Symbol='*') or (Symbol='/');
[KV2] 12Ø8:     Writeln(Symbol);
[8L2] 12Ø9:     Write('Zahl2 : ');
[682] 121Ø:     Readln(ZahlB);
[1U4] 1211:     if ZahlB = '' then Zahl2 := Zahl2
[ØØ4] 1212:       else Val(ZahlB, Zahl2, i);
[CG4] 1213:     if (Symbol='/') and (Zahl2 = Ø)
[X92] 1214:       then Zahl2 := 1;
[YV1] 1215:     GotoXY(9,3);
[H53] 1216:     Writeln(Zahl2:1:11);
[LV1] 1217:     CursorOff;
[XF1] 1218:     Writeln;
[NF2] 1219:     case Symbol of
[EL4] 122Ø:     '+' : Ergebnis := Zahl1 + Zahl2;
[SL4] 1221:     '-' : Ergebnis := Zahl1 - Zahl2;
[9L4] 1222:     '*' : Ergebnis := Zahl1 * Zahl2;
[2M4] 1223:     '/' : Ergebnis := Zahl1 / Zahl2;
[ZNØ] 1224:     end;
[1U5] 1225:     Writeln('Ergebnis: ', Ergebnis:1:11);
[XF1] 1226:     Writeln;
[WT5] 1227:     Writeln('Noch eine Rechnung? (j/n)');
[V74] 1228:     Weiter := UpCase(ReadKey);
[N23] 1229:   until Weiter = 'N';
[FL2] 123Ø:   Auswahl_zeigen
[ZNØ] 1231: end;
[ØØØ] 1232:
[ZWØ] 1233: {$F+}
[DB4] 1234: PROCEDURE Hauptprogramm;
[MUØ] 1235: begin
[XA2] 1236:   AltX:= WhereX;
[ØB2] 1237:   AltY:= WhereY;
[ZH3] 1238:   Bildschirm_sichern;
[LV1] 1239:   CursorOff;
[F11] 124Ø:   Maske;
[VU2] 1241:   Auswahl_zeigen;
[VN1] 1242:   Seite := 1;
[DH2] 1243:   CursorPos := 1;
[X93] 1244:   Bildschirm_merken;
[DX1] 1245:   Inh := true;
[MK3] 1246:   Laufwerk_bestimmen;
[FR3] 1247:   Bildschirm_erinnern;
[P31] 1248:   repeat
[F52] 1249:     Auswahl := #1;
[6D2] 125Ø:     Auswahl1 := #1;
[4N1] 1251:     Wahl := #1;
[FE1] 1252:     Anzeige;
[P31] 1253:     repeat
[P31] 1254:       repeat
[P31] 1255:         repeat
[V11] 1256:           Datum;
[VRØ] 1257:           Zeit
[K13] 1258:         until KeyPressed;
[NF4] 1259:           Auswahl := UpCase(ReadKey);
[Z83] 126Ø:           until (Auswahl=#Ø) or
[FD4] 1261:           (Auswahl=#13) or (Auswahl=#27)
[T45] 1262:           or (Auswahl='A') or (Auswahl='B')
[755] 1263:           or (Auswahl='C') or (Auswahl='D')
[N55] 1264:           or (Auswahl='E') or (Auswahl='F')
[Z65] 1265:           or (Auswahl='K') or (Auswahl='L')
[595] 1266:           or (Auswahl='U') or (Auswahl='V')
[G85] 1267:           or (Auswahl='R') or (Auswahl='S')
[4H2] 1268:           or (Auswahl='N')
[CN2] 1269:           or (Auswahl='T');
[6N2] 127Ø:           case Auswahl of
[N42] 1271:             #Ø     : Cursor;
[PW2] 1272:             #13    : Ausfuehren;
[CA4] 1273:             'A'    : Laufwerk_wechseln;
[FA4] 1274:             'B'    : Laufwerk_wechseln;
[KA4] 1275:             'C'    : Laufwerk_wechseln;
[NA4] 1276:             'D'    : Laufwerk_wechseln;
[SA4] 1277:             'E'    : Laufwerk_wechseln;
[VA4] 1278:             'F'    : Laufwerk_wechseln;
[Z12] 1279:             'K'    : Kopie;
[CY2] 128Ø:             'U'    : Umbenennen;
[EG2] 1281:             'L'    : Loeschen;
[Z23] 1282:             'V'    : Verzeichnis;
[6U2] 1283:             'R'    : Entfernen;
[2S2] 1284:             'S'    : Anschauen;
[F63] 1285:             'N'    : Notizzettel;
[8L3] 1286:             'T'    : Taschenrechner;
[ZNØ] 1287:           end;
[GM5] 1288:       until (Auswahl='A') or (Auswahl='B')
[755] 1289:       or (Auswahl='C') or (Auswahl='D')
[N55] 129Ø:       or (Auswahl='E') or (Auswahl='F')
[YL2] 1291:       or (Auswahl=#27);
[M33] 1292:     until Auswahl = #27;
[DB1] 1293:     Invoff;
[GH2] 1294:     Window(1,1,8Ø,25);
[Z83] 1295:     Bildschirm_zeigen;
[MN1] 1296:     CursorOn;
[873] 1297:     GotoXY(AltX,AltY);
[ZNØ] 1298: end;
[1XØ] 1299: {$F-}
[ØØØ] 13ØØ:
[XS2] 13Ø1: PROCEDURE Hilfe;
[MUØ] 13Ø2: begin
[XF1] 13Ø3:   Writeln;
[CZ4] 13Ø4:   Writeln(' TSR-Datei-Manager ',
[P92] 13Ø5:     'Version 1.Ø3');
[AYA] 13Ø6:   Writeln('──────────────────',
[DP6] 13Ø7:     '──────────');
[KE5] 13Ø8:   Writeln('Aufruf mit folgenden ',
[CG2] 13Ø9:     'Parametern:');
[XF1] 131Ø:   Writeln;
[AL5] 1311:   Writeln('DMANAGER <Grafikkarte> ',
[3Y5] 1312:     '<Anzahl der Diskettenlaufwerke>');
[XF1] 1313:   Writeln;
[4A6] 1314:   Writeln(' C oder c = Farbgrafikkarte');
[D36] 1315:   Writeln(' H oder h = Hercules-Karte');
[XF1] 1316:   Writeln;
[1S5] 1317:   Writeln(' 1 = Diskettenlaufwerk B ',
[N33] 1318:     'nicht vorhanden');
[5S5] 1319:   Writeln(' 2 = Diskettenlaufwerk B ',
[W22] 132Ø:     'vorhanden');
[XF1] 1321:   Writeln;
[6A6] 1322:   Writeln('Bei beiden Parametern wird ',
[4G2] 1323:     'nur jeweils');
[2M4] 1324:   Writeln('das erste Zeichen ',
[KA3] 1325:     'berücksichtigt!');
[XF1] 1326:   Writeln;
[DUØ] 1327:   Halt;
[ZNØ] 1328: end;
[ØØØ] 1329:
[MUØ] 133Ø: begin
[4F4] 1331:   Parameter1a := ParamStr(1);
[V36] 1332:   Parameter1 := UpCase(Parameter1a[1]);
[9F4] 1333:   Parameter2a := ParamStr(2);
[MT4] 1334:   Parameter2 := Parameter2a[1];
[SY3] 1335:   if (Parameter1 <> 'C') then
[TØ5] 1336:     if (Parameter1 <> 'H') then Hilfe;
[9Y3] 1337:   if (Parameter2 <> '1') then
[6Ø5] 1338:     if (Parameter2 <> '2') then Hilfe;
[FCD] 1339:   Writeln('┌──────────────────────────',
[YAE] 134Ø:       '──────────────────────┐');
[HU5] 1341:   Writeln('| (C) DOS International  ',
[UX5] 1342:       '  Autor: Siegfried Stößlein |');
[AAD] 1343:   Writeln('├──────────────────────────',
[X9E] 1344:       '──────────────────────┤');
[AX6] 1345:   Writeln('|  TSR-Datei-Manager Version ',
[R64] 1346:       '1.Ø3 ist installiert.  |');
[7F6] 1347:   Writeln('|  ***  Aufruf mit [Alt-F1Ø] ',
[CQ4] 1348:       'oder [Ctrl-F1Ø].  ***  |');
[SPF] 1349:   Writeln('└──────────────────────────',
[WXB] 135Ø:       '──────────────────────┘');
[YR2] 1351:   TurboSS := Sseg;
[E23] 1352:   TurboSP := Sptr+4;
[UG4] 1353:   IntProg := @Hauptprogramm;
[WS3] 1354:   GetIntVec($16,altint16);
[GE3] 1355:   SetIntVec($16,@Int16);
[634] 1356:   SetIntVec($1b,SaveInt1b);
[PZ3] 1357:   SetIntVec(Ø,SaveIntØØ);
[841] 1358:   Keep(Ø);
[KNØ] 1359: end.
```

Listing. Der TSR-Dateimanager »dmanager.pas« bewältigt die üblichen Dateiverwaltungsfunktionen

SAM III, Version 1.1

Der in der DOS Extra 8 abgedruckte Makro-Assembler SAM III stieß bei den DOS-Lesern auf reges Interesse. Allerdings offenbarten sich bei der täglichen Arbeit mit dem Assembler auch einige Fehlfunktionen, die nun in der neuen Version 1.1 behoben sind.

Mehrere Leser, die mit dem Makro-Assembler SAM III aus DOS Extra 8 arbeiten, haben uns geschrieben und auf folgende fünf Fehlfunktionen hingewiesen:

1.) Indexaddressierte Befehle wie zum Beispiel

```
mov -12[BP],AX)
```

werden zwar richtig übersetzt, doch die Adreßumrechnung der nachfolgenden Sprungsymbole stimmt nicht mehr, wenn der Index-Offset zwischen -128 und -1 oder zwischen + 128 und + 255 liegt.

2.) Die Multiplikations- und Divisionsbefehle »mul«, »imul«, »div« und »idiv« werden fehlerhaft assembliert, wenn 8-Bit-Operationen (zum Beispiel »mul AX,DL«) durchgeführt werden sollen.

3.) Der Pseudobefehl »even« paßt bei ungerader Adresse den Codezeiger zwar richtig an, fügt aber nicht den erforderlichen Nop-Befehl ein.

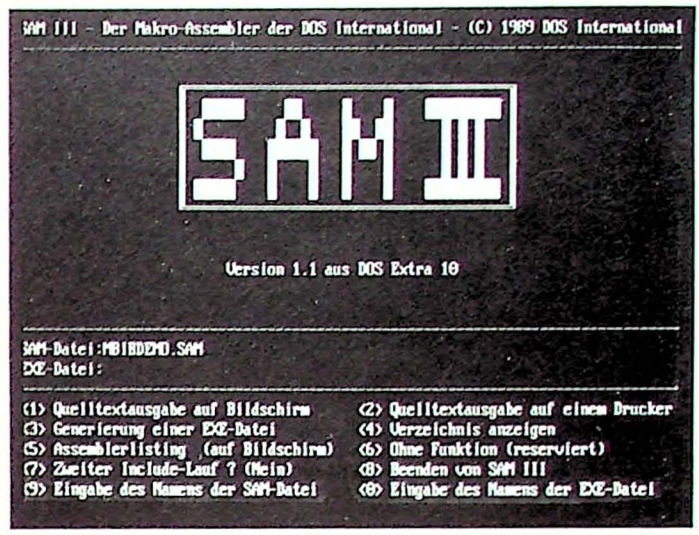
4.) Der Pseudobefehl »ds« reserviert zwar den angeforderten Speicherplatz an der richtigen Stelle, verkürzt aber den zu generierenden Code um die Anzahl der reservierten Bytes.

5.) Die Pseudobefehle »segment code« und »segment stack« werden nicht akzeptiert, wenn in der gleichen Zeile ein Kommentar folgt.

Diese Fehler wurden nun in der neuen Version 1.1 von SAM III behoben. Das Stack-Segment muß sich nun allerdings im Gegensatz zu den in DOS Extra 8 abgedruckten Listings vor dem Codesegment befinden. Darüber hinaus lassen sich an SAM III nun auch in der Quick-Basic-Ausführung Kommandozeilenparameter übergeben.

Listing 1 zeigt die neue Version von SAM III. Sämtliche Veränderungen im Quelltext der Version 1.1 gegenüber der Version 1.0 sind durch die Zeichenfolge »***« gekennzeichnet.

Des weiteren darf ein Makro, das eine Sprungmarke enthält, nicht mehrfach im Programm verwendet werden, weil SAM III in diesem Fall eine Mehrfachdefinition der Sprungmarken bemängelt. Sprungbefehle innerhalb eines Makros müssen daher recht umständlich durch DB-Befehle ersetzt werden. Dieses Verfahren demonstrieren die beiden Beispielprogramme »mbibdemo.sam« (Listing 2) und



Der Makro-Assembler SAM III präsentiert sich in der Version 1.1

»macrobib.sam« (Listing 3), die bei gesetztem zweiten Include-Lauf assembliert werden müssen.

Dieses Problem läßt sich vermutlich nur durch die zusätzliche Implementierung von lokalen Marken lösen. Denkbar ist vielleicht auch, Makros getrennt zu compilieren und anschließend an der aufzurufenden Stelle einzubinden. Vielleicht kann ein DOS-Leser zu diesem Problem eine Lösung anbieten?

Die Version 1.1 des Makro-Assembler SAM III wurde in einigen wichtigen Punkten verbessert

In vielen Assembler-Listings sind die Assemblerbefehle »ja«, »jae«, »jb«, »jbe«, »jc«, »je«, »jg«, »jge«, »jl«, »jle«, »jnc«, »jne«, »jnp«, »jp«, »shl«, »shlb« und »shlw« enthalten, die von SAM III nicht akzeptiert werden. Deshalb wurde die SAM-III-Datei »sam3.dat« um diese Befehle ergänzt und zur besseren Unterscheidung in »sam3v11.dat« umbenannt (Listing 4). Dementsprechend wurde auch der Aufruf der Datei im Quellcode von SAM III geändert. Zusätzlich wurden die Variablen »befehlzahl«, »vordef« und »symbolzahl« aus der Datei »sam3v11.dat« entfernt und im Quelltext von SAM III durch die Konstanten 231, 29 und 500 ersetzt, weil eine dynamische Dimensionierung in der Art

```
b=10: DIM a$(b)
```

unter SBasic nicht möglich ist. Wer dagegen SAM III mit Quick Basic oder Turbo Basic compiliert, braucht diese Änderungen gegenüber der Version 1.0 nicht durchzuführen.

(Walter Hermanns/ma)

Name	sam3.asc
Sprache	SBasic, Quick Basic, Turbo Basic
Grafikkarte	jede
Besonderheiten	Zum Compilieren mit SBasic ist der Präprozessor aus DOS Extra 7 erforderlich

```
[ZW5]    1: 'Makro-Assembler SAM III, Version 1.1
[4W6]    2: '(C) 1989 Redaktionsbüro Everts&Hagedorn
[5P5]    3: '          Redaktion DOS International
[M16]    4: '          Geschrieben von Markus Zietlow
[166]    5: '          Veröffentlicht in DOS Extra 8
[RH2]    6: DEFINT A-Z: CLS
[GME]    7: '*** Alle Änderungen gegenüber Version 1.Ø
            sind durch die Zeichenfolge *** gekennzeich
            net
[E2Ø]    8: '
[F2B]    9: 'Überprüfen, ob in der Kommandozeile Parame
            ter übergeben wurden
[E2Ø]   1Ø: '
[C3E]   11: 'Die folgenden drei Programmzeilen sind nur
            für die Quick-Basic-Version gültig:
```

Listing 1. »sam3.asc« ist die neue Version 1.1 von SAM III

```
[H89]   12: quelldat$ = COMMAND$: i = INSTR(COMMAND$, "
            "): ausgabe$ = " "
[6AB]   13: IF i > Ø THEN quelldat$ = LEFT$(COMMAND$, i
            - 1): zieldat$ = MID$(COMMAND$, i)
[BR9]   14: WHILE LEFT$(zieldat$, 1) = " ": zieldat$ =
            MID$(zieldat$, 2): WEND
[ØØØ]   15:
[PSB]   16: 'Die folgende Programmzeile ist nur für die
            SBasic-Version gültig:
[5C9]   17: quelldat$ = COMMAND$(1): zieldat$ = COMMAND
            $(2): ausgabe$ = " "
[ØØØ]   18:
[6L5]   19: 'Bearbeiten des Quelltextnamens
[5DC]   2Ø: IF quelldat$ = "-" THEN ausgabe$ = "-": que
            lldat$ = "": zieldat$ = "": GOTO pruefen
[617]   21: IF zieldat$ = "-" THEN ausgabe$ = "-": ziel
            dat$ = ""
[WF1]   22: pruefen:
[HW2]   23: pruef$ = quelldat$
[E2Ø]   24: '
[H57]   25: 'Feststellen, ob Erweiterung angegeben ist
[GE4]   26: WHILE INSTR(pruef$, "\") <> Ø
[UL6]   27:   pruef$ = MID$(pruef$, INSTR(pruef$, "\")
            + 1)
[TRØ]   28: WEND
[E2Ø]   29: '
[HNB]   3Ø: IF (INSTR(pruef$, ".") = Ø) AND (pruef$ <>
            "") THEN quelldat$ = quelldat$ + ".SAM"
[KR4]   31: IF zieldat$ = "" THEN GOTO init
[E2Ø]   32: '
[YG5]   33: 'Bearbeiten des Zieldateinamens
[Ø51]   34: DOS = -1
[ZM2]   35: pruef$ = zieldat$
[E2Ø]   36: '
[H57]   37: 'Feststellen, ob Erweiterung angegeben ist
[GE4]   38: WHILE INSTR(pruef$, "\") <> Ø
[UL6]   39:   pruef$ = MID$(pruef$, INSTR(pruef$, "\")
            + 1)
[TRØ]   4Ø: WEND
[XUØ]   41: '*** ->
[2BB]   42: IF (INSTR(pruef$, ".") = Ø) AND (pruef$ <>
            "") THEN zieldat$ = zieldat$ + ".EXE"
[GUØ]   43: '<- ***
[LW2]   44: 'Initialisierung
[8VØ]   45: init:
[A26]   46: 'Assemblerlisting auf den Bildschirm
[EH1]   47: asslis = 1
[2P4]   48: IF ausgabe$ = "-" THEN asslis = 4
[RB3]   49: 'Keine Symboltabelle
[PK1]   5Ø: symaus = Ø
[FU4]   51: 'Keine Cross-Referenztabelle
[5C1]   52: cross = Ø
[ZT3]   53: 'Datenzeilen pro Seite
[VL1]   54: ln.max = 72
[K23]   55: 'Zeichen pro Zeile
[EL1]   56: br.max = 8Ø
[PX2]   57: wahr = -1: falsch = Ø
[G22]   58: trenn$ = " ," + CHR$(9)
[ØH2]   59: makros = falsch
[E2Ø]   6Ø: '
[BU4]   61: 'Definition der Datenfelder
[LY1]   62: DIM param$(8)
[6N5]   63: maxbefehle = 255: stapelgroesse = 1Ø
[2F6]   64: DIM befehltab(255): DIM prozedurtyp(1Ø)
[E2Ø]   65: '
[G25]   66: '*** -> Einlesen der Datei 'SAM3V11.DAT'
[F65]   67: OPEN "sam3v11.dat" FOR INPUT AS #3
[PY4]   68: IF EOF(3) THEN GOTO samdat.fehl
[EV7]   69: DIM kommando$(231), operwert(231), quellreg
            typ(231)
[9A5]   7Ø: DIM zielregtyp(231), flag(231) '<- ***
[V82]   71: FOR i = 1 TO 231
[M2B]   72:   INPUT #3, kommando$(i), operwert(i), ziel
            regtyp(i), quellregtyp(i), flag(i)
[U11]   73: NEXT i
[E2Ø]   74: '
[579]   75: DIM symbol$(5ØØ), wert1(5ØØ), wert2(29), sy
            mboltyp(5ØØ) '<- ***
[282]   76: FOR i = 1 TO 29
[VC8]   77:   INPUT #3, symbol$(i), wert1(i), wert2(i),
            symboltyp(i)
[U11]   78: NEXT i
[H82]   79: aktsymbol = 29
[251]   8Ø: CLOSE #3
[BA4]   81: IF DOS THEN GOTO assembler
[E2Ø]   82: '
[MG7]   83: 'Titelbild, Ausgabe der Benutzeroberfläche
[Ø94]   84: ' und Copyright-Meldungen
[E11]   85: menue:
[PNØ]   86: CLS
[AZD]   87: PRINT "SAM III - Der Makro-Assembler der DO
            S International - (C) 1989 DOS Internationa
            l";
[PC3]   88: PRINT STRING$(8Ø, "-");
[B85]   89: LOCATE 16, 1: PRINT STRING$(8Ø, "-");
[1S2]   9Ø: PRINT "SAM-Datei:"
[9S2]   91: PRINT "EXE-Datei:"
[PC3]   92: PRINT STRING$(8Ø, "-");
[C3E]   93: PRINT "<1> Quelltextausgabe auf Bildschirm
                <2> Quelltextausgabe auf einem Drucker
            "
[D3B]   94: PRINT "<3> Generierung einer EXE-Datei
                <4> Verzeichnis anzeigen"
[98B]   95: PRINT "<5> Assemblerlisting  (
            )    <6> Ohne Funktion (reserviert)"
[XSA]   96: PRINT "<7> Zweiter Include-Lauf ? (Nein)
                <8> Beenden von SAM III"
[A1D]   97: PRINT "<9> Eingabe des Namens der SAM-Datei
                <Ø> Eingabe des Namens der EXE-Datei";
[E2Ø]   98: '
[Y21]   99: titel:
[TZ7]  1ØØ: LOCATE 4, 2Ø: PRINT CHR$(201); STRING$(39,
            2Ø5); CHR$(187)
[ACH]  1Ø1: LOCATE 5, 2Ø: PRINT CHR$(186); "  █████
              ██     █    █    █████████  "; CHR$(186)
[DMC]  1Ø2: LOCATE 6, 2Ø: PRINT CHR$(186); " █
            █  █     ██   ██       █ █ █  "; CHR$(186)
[5FG]  1Ø3: LOCATE 7, 2Ø: PRINT CHR$(186); "  █████   █
            █████    █ ██ █       █ █ █  "; CHR$(186)
[T8C]  1Ø4: LOCATE 8, 2Ø: PRINT CHR$(186); "       █  █
                 █   █    █       █ █ █  "; CHR$(186)
[L6H]  1Ø5: LOCATE 9, 2Ø: PRINT CHR$(186); " █████    █
                 █   █    █    █████████  "; CHR$(186)
[848]  1Ø6: LOCATE 1Ø, 2Ø: PRINT CHR$(2ØØ); STRING$(39,
             2Ø5); CHR$(188)
[UF7]  1Ø7: LOCATE 13, 26: PRINT "Version 1.1 aus DOS E
            xtra 1Ø"
[E2Ø]  1Ø8: '
[5C1]  1Ø9: status:
[VM9]  11Ø: LOCATE 17, 11: PRINT quelldat$; STRING$(7Ø
            - LEN(quelldat$), 32);
[789]  111: LOCATE 18, 11: PRINT zieldat$; STRING$(7Ø -
            LEN(zieldat$), 32);
[YDV]  112: LOCATE 22, 24: IF asslis = 1 THEN PRINT "au
            f Bildschirm" ELSE IF asslis = 2 THEN PRINT
             "auf Drucker   " ELSE IF asslis = 3 THEN P
            RINT "in eine Datei " ELSE PRINT "nicht aus
            geben"
[E2Ø]  113: '
[3G6]  114: 'Eingabe des gewünschten Menüpunktes
[UB1]  115: eingabe:
[RS5]  116: a$ = INKEY$: IF a$ = "" THEN GOTO eingabe
[TØ2]  117: a = ASC(a$) - 48
[VF5]  118: IF (a < Ø) OR (a > 9) THEN GOTO eingabe
[E2Ø]  119: '
[BE7]  12Ø: 'Ausgabe des Quelltextes auf den Bildschirm
[7L4]  121: IF a = 1 THEN GOTO aus.monitor
[E2Ø]  122: '
[UT6]  123: 'Ausgabe des Quelltextes auf den Drucker
[5K4]  124: IF a = 2 THEN GOTO aus.drucker
[E2Ø]  125: '
[EZ6]  126: 'Assemblieren des angegebenen Quelltextes
[V74]  127: IF a = 3 THEN GOTO assembler
[E2Ø]  128: '
[L56]  129: 'Anzeigen des Inhaltsverzeichnisses
[VG4]  13Ø: IF a <> 4 THEN GOTO sam.finis
[SL1]  131: LOCATE 3, 1
[772]  132: FOR i = 1 TO 13
[R93]  133:   PRINT STRING$(8Ø, 32);
[U11]  134: NEXT i
[SL1]  135: LOCATE 3, 1
[KK2]  136: FILES "*.??m" '<- ***
[6P5]  137: PRINT "- Bitte eine Taste drücken -"
[5H3]  138: WHILE INKEY$ = "": WEND
[SL1]  139: LOCATE 3, 1
[772]  14Ø: FOR i = 1 TO 13
[R93]  141:   PRINT STRING$(8Ø, 32);
[U11]  142: NEXT i
[WT1]  143: GOTO titel
[E2Ø]  144: '
[N83]  145: 'Beenden von SAM III
[7S1]  146: sam.finis:
[G1B]  147: IF a = 8 THEN CLS : PRINT "SAM III hat sein
            e Arbeit beendet.": PRINT : END
[E2Ø]  148: '
[ZS4]  149: 'Form des Assembler-Listings
[545]  15Ø: IF a <> 5 THEN GOTO quell.eingabe
[P23]  151: asslis = asslis + 1
[2C4]  152: IF asslis = 5 THEN asslis = 1
[R62]  153: GOTO status
[E2Ø]  154: '
[PW4]  155: 'Eingabe des Quelltextnamens
[FB2]  156: quell.eingabe:
[PZ4]  157: IF a <> 9 THEN GOTO ziel.eingabe
[SL1]  158: LOCATE 3, 1
[772]  159: FOR i = 1 TO 13
[R93]  16Ø:   PRINT STRING$(8Ø, 32);
[U11]  161: NEXT i
[9HC]  162: LOCATE 8, 1: LINE INPUT "Geben Sie bitte de
            n Namen der SAM-Datei ein: "; quelldat$
[XL1]  163: LOCATE 8, 1
[M12]  164: FOR q = 1 TO 4
[R93]  165:   PRINT STRING$(8Ø, 32);
[K21]  166: NEXT q
[4X4]  167: quelldat$ = LEFT$(quelldat$, 66)
[HW2]  168: pruef$ = quelldat$
[GE4]  169: WHILE INSTR(pruef$, "\") <> Ø
[UL6]  17Ø:   pruef$ = MID$(pruef$, INSTR(pruef$, "\")
```

```
            + 1)
 [6H1]  171: WEND '*** ->
 [HNB]  172: IF (INSTR(pruef$, ".") = Ø) AND (pruef$ <>
             "") THEN quelldat$ = quelldat$ + ".SAM"
 [YK2]  173: GOTO titel '<- ***
 [E2Ø]  174: '
 [5R4]  175: 'Eingabe des Zieldateinamens
 [Y42]  176: ziel.eingabe:
 [L64]  177: IF a <> Ø THEN GOTO in.lauf
 [SL1]  178: LOCATE 3, 1
 [772]  179: FOR i = 1 TO 13
 [R93]  18Ø:   PRINT STRING$(8Ø, 32);
 [U11]  181: NEXT i
 [Z9C]  182: LOCATE 8, 1: LINE INPUT "Geben Sie bitte de
             n Namen der EXE-Datei ein: "; zieldat$
 [XL1]  183: LOCATE 8, 1
 [M12]  184: FOR q = 1 TO 4
 [R93]  185:   PRINT STRING$(8Ø, 32);
 [K21]  186: NEXT q
 [DH4]  187: zieldat$ = LEFT$(zieldat$, 66)
 [ZM2]  188: pruef$ = zieldat$
 [GE4]  189: WHILE INSTR(pruef$, "\") <> Ø
 [UL6]  19Ø:   pruef$ = MID$(pruef$, INSTR(pruef$, "\")
             + 1)
 [6H1]  191: WEND '*** ->
 [2BB]  192: IF (INSTR(pruef$, ".") = Ø) AND (pruef$ <>
             "") THEN zieldat$ = zieldat$ + ".EXE"
 [YK2]  193: GOTO titel '<- ***
 [E2Ø]  194: '
 [D36]  195: 'Zweiter Include-Lauf zum Einbinden
 [244]  196: ' externer Makros nötig ?
 [LA1]  197: in.lauf:
 [884]  198: IF a <> 7 THEN GOTO eingabe
 [PFG]  199: IF makros THEN makros = falsch: LOCATE 23,
             29: PRINT "Nein"; :  ELSE makros = wahr: LO
             CATE 23, 29: PRINT " Ja ";
 [T12]  2ØØ: GOTO eingabe
 [E2Ø]  2Ø1: '
 [3K5]  2Ø2: 'Ausgabe eines Quelltextes auf den
 [KZ1]  2Ø3: ' Bildschirm
 [U82]  2Ø4: aus.monitor:
 [D77]  2Ø5: IF quelldat$ <> "" THEN GOTO monitor.con '*
             ** ->
 [7WE]  2Ø6: LOCATE 15, 4: PRINT "Sie haben noch keine S
             AM-Datei angegeben. Drücken Sie bitte eine
             Taste !"
 [5H3]  2Ø7: WHILE INKEY$ = "": WEND
 [716]  2Ø8: LOCATE 15, 1: PRINT STRING$(8Ø, 32) '<- ***
 [LZ6]  2Ø9: GOTO eingabe: 'Zurück ins Hauptprogramm
 [E2Ø]  21Ø: '
 [S72]  211: monitor.con:
 [PNØ]  212: CLS
 [WT4]  213: OPEN quelldat$ FOR INPUT AS #1
 [MF1]  214: zeilen = Ø
 [3E2]  215: WHILE NOT EOF(1)
 [VB3]  216:   LINE INPUT #1, PZEILE$
 [8Ø3]  217:   zeilen = zeilen + 1
 [TF5]  218:   PRINT zeilen; : PRINT ": "; PZEILE$
 [TRØ]  219: WEND
 [3WØ]  22Ø: CLOSE
 [6YD]  221: PRINT : PRINT "Die Ausgabe der "; zeilen; :
              PRINT " Zeile"; : IF zeilen <> 1 THEN PRIN
             T "n";
 [TFL]  222: PRINT " der SAM-Datei auf dem Bildschirm is
             t beendet.": PRINT "                Drücken
              Sie bitte eine Taste, um fortzufahren.";
 [5H3]  223: WHILE INKEY$ = "": WEND
 [WS1]  224: GOTO menue
 [E2Ø]  225: '
 [UT6]  226: 'Ausgabe des Quelltextes auf den Drucker
 [R72]  227: aus.drucker:
 [P47]  228: IF quelldat$ <> "" THEN GOTO drucker.con '*
             ** ->
 [7WE]  229: LOCATE 15, 4: PRINT "Sie haben noch keine S
             AM-Datei angegeben. Drücken Sie bitte eine
             Taste !"
 [5H3]  23Ø: WHILE INKEY$ = "": WEND
 [716]  231: LOCATE 15, 1: PRINT STRING$(8Ø, 32) '<- ***
 [LZ6]  232: GOTO eingabe: 'Zurück ins Hauptprogramm
 [E2Ø]  233: '
 [N62]  234: drucker.con:
 [PNØ]  235: CLS
 [WT4]  236: OPEN quelldat$ FOR INPUT AS #1
 [P86]  237: zeilen = Ø: Seite = 1: AZEILEN = Ø: flag =
             Ø
 [E2Ø]  238: '
 [R16]  239: WHILE NOT EOF(1): 'Schließe Dateien
 [VB3]  24Ø:   LINE INPUT #1, PZEILE$
 [PH1]  241: druck.w1:
 [W75]  242:   IF AZEILEN <> Ø THEN GOTO druck.w2
 [X66]  243:   A1 = INT((br.max - 8Ø) / 2): A2 = br.max
             - 8Ø - A1
 [UNU]  244:   LPRINT : LPRINT "Listing einer SAM-Datei
                        "; STRING$(A1, 32); "Datum: ";
              MID$(DATE$, 4, 2); "."; LEFT$(DATE$, 2); "
             ."; RIGHT$(DATE$, 4); "                "; S
             TRING$(A2, 32); "Seite: "; Seite
 [7F6]  245:   LPRINT : LPRINT "SAM-Datei:"; quelldat$
 [HD7]  246:   LPRINT STRING$(br.max, "-"): LPRINT : AZE
             ILEN = 6
 [RH1]  247: druck.w2:
 [PL4]  248:   IF flag = Ø THEN GOTO druck.w3
 [PW7]  249:   LPRINT "        "; LEFT$(PZEILE$, 73): zei
             len = zeilen + 1
 [152]  25Ø:   GOTO druck.w4
 [SH1]  251: druck.w3:
 [8Ø3]  252:   zeilen = zeilen + 1
 [N97]  253:   LPRINT zeilen; : LPRINT ": "; LEFT$(PZEIL
             E$, 73)
 [TH1]  254: druck.w4:
 [V1B]  255:   IF LEN(PZEILE$) > 73 THEN flag = -1: PZEI
             LE$ = MID$(PZEILE$, 74) ELSE flag = Ø
 [873]  256:   AZEILEN = AZEILEN + 1
 [Ø7D]  257:   IF AZEILEN = ln.max - 3 THEN LPRINT : LPR
             INT : LPRINT : AZEILEN = Ø: Seite = Seite +
              1
 [P54]  258:   IF flag THEN GOTO druck.w1
 [TRØ]  259: WEND
 [E2Ø]  26Ø: '
 [3WØ]  261: CLOSE
 [4L3]  262: FOR q = 1 TO 72 - zeilen
 [H51]  263:   LPRINT
 [K21]  264: NEXT q
 [3AF]  265: PRINT : PRINT : PRINT "   Die Ausgabe der "
             ; zeilen; : PRINT " Zeile"; : IF zeilen <>
             1 THEN PRINT "n";
 [WVL]  266: PRINT " der SAM-Datei auf dem Drucker ist b
             eendet.": PRINT : PRINT "                D
             rücken Sie bitte eine Taste, um fortzufahre
             n.";
 [5H3]  267: WHILE INKEY$ = "": WEND
 [WS1]  268: GOTO menue
 [E2Ø]  269: '
 [U84]  27Ø: 'Erzeugen einer EXE-Datei
 [LT1]  271: assembler:
 [Ø45]  272: IF DOS THEN GOTO Hauptprogramm
 [FT6]  273: IF quelldat$ <> "" THEN GOTO assem.con '***
             ->
 [7WE]  274: LOCATE 15, 4: PRINT "Sie haben noch keine S
             AM-Datei angegeben. Drücken Sie bitte eine
             Taste !"
 [5H3]  275: WHILE INKEY$ = "": WEND
 [716]  276: LOCATE 15, 1: PRINT STRING$(8Ø, 32) '<- ***
 [LZ6]  277: GOTO eingabe: 'Zurück ins Hauptprogramm
 [E2Ø]  278: '
 [NP1]  279: assem.con:
 [RG6]  28Ø: IF zieldat$ <> "" THEN GOTO Hauptprogramm
 [E2Ø]  281: '
 [A93]  282: zieldat$ = quelldat$
 [VP2]  283: P = LEN(zieldat$)
 [9E4]  284: WHILE MID$(zieldat$, P, 1) <> "."
 [A21]  285:   P = P - 1
 [TRØ]  286: WEND
 [UE5]  287: zieldat$ = LEFT$(zieldat$, P) + "exe"
 [P93]  288: GOTO Hauptprogramm
 [E2Ø]  289: '
 [8Y6]  29Ø: 'Zweiter Lauf (verarbeitet INCLUDE-Dateien
 [L67]  291: ' und fügt eventuell vorhandene MAKROS ein)
 [ØU1]  292: lauf.zwei:
 [ZCA]  293: OPEN "work5.wrk" FOR INPUT AS #1: OPEN "wor
             k2.wrk" FOR OUTPUT AS #4
 [W41]  294: lauf = Ø
 [H82]  295: aktsymbol = 29
 [XU1]  296: codezeig = Ø
 [WV1]  297: zeilnum = Ø
 [3E2]  298: WHILE NOT EOF(1)
 [SL6]  299:   GOSUB lies.zeile: 'Einlesen einer Zeile
 [YR5]  3ØØ:   GOSUB analyse: 'Zeile übersetzen
 [E2Ø]  3Ø1:   '
 [B16]  3Ø2:   'Falls INCLUDE-Anweisungen vorhanden
 [1M6]  3Ø3:   ' sind, werden sie, so weit als möglich,
 [442]  3Ø4:   ' ausgeführt
 [6H6]  3Ø5:   IF befehl$ <> "INCLUDE" THEN GOTO do.befe
             hl
 [277]  3Ø6:   datei$ = MID$(zeile$, INSTR(zeile$, befeh
             l$) + 8)
 [SN3]  3Ø7:   WHILE INSTR(datei$, ";")
 [4B6]  3Ø8:     datei$ = LEFT$(datei$, INSTR(datei$, ";
             ") - 1)
 [TRØ]  3Ø9:   WEND
 [5T3]  31Ø:   WHILE LEFT$(datei$, 1) = " "
 [3D3]  311:     datei$ = MID$(datei$, 2)
 [TRØ]  312:   WEND
 [X14]  313:   WHILE RIGHT$(datei$, 1) = " "
 [YC5]  314:     datei$ = LEFT$(datei$, LEN(datei$) - 1)
 [TRØ]  315:   WEND
 [R94]  316:   OPEN datei$ FOR INPUT AS #5
 [E25]  317:   IF EOF(5) THEN GOTO fehl.include
 [E2Ø]  318:   '
 [LE2]  319:   WHILE NOT EOF(5)
 [LE2]  32Ø:     LINE INPUT #5, a$
 [GN1]  321:     PRINT #4, a$
 [TRØ]  322:   WEND
 [451]  323:   CLOSE #5
 [ME1]  324:   zeile$ = ""
```

Listing 1. »sam3.asc« ist die neue Version 1.1 von SAM III

```
[EE2]  325:   GOTO m.schluss
[E2Ø]  326:   '
[U46]  327:   'Falls DO-Anweisungen vorhanden sind,
[CL5]  328:   ' werden sie so weit als möglich
[442]  329:   ' ausgeführt
[DL1]  33Ø: do.befehl:
[3N5]  331:   IF befehl$ <> "DO" THEN GOTO m.schluss
[TZ7]  332:   Parameter$ = MID$(zeile$, INSTR(zeile$, b
              efehl$) + 3)
[ME4]  333:   WHILE INSTR(Parameter$, ";")
[ML8]  334:     Parameter$ = LEFT$(Parameter$, INSTR(Pa
              rameter$, ";") - 1)
[TRØ]  335:   WEND
[1K4]  336:   WHILE LEFT$(Parameter$, 1) = " "
[LZ4]  337:     Parameter$ = MID$(Parameter$, 2)
[TRØ]  338:   WEND
[BU4]  339:   WHILE RIGHT$(Parameter$, 1) = " "
[6M7]  34Ø:     Parameter$ = LEFT$(Parameter$, LEN(Para
              meter$) - 1)
[TRØ]  341:   WEND
[E43]  342:   FOR zaehler = 1 TO 8
[HV2]  343:     param$(zaehler) = ""
[G42]  344:   NEXT zaehler
[E2Ø]  345:   '
[PD1]  346:   MNAME$ = ""
[6HD]  347:   IF INSTR(Parameter$, ",") = Ø THEN MNAME$
               = Parameter$: Parameter$ = "": GOTO m.para
              meter
[9S7]  348:   MNAME$ = LEFT$(Parameter$, INSTR(Parameter$, ",") - 1)
[DM8]  349:   Parameter$ = MID$(Parameter$, INSTR(Param
              eter$, ",") + 1)
[E2Ø]  35Ø:   '
[Y32]  351: m.parameter:
[5HH]  352:   IF MNAME$ = "" THEN PRINT : PRINT "Beim A
              ufruf des Makros fehlt die Angabe des Makro
              namens.": GOTO macro.fehler
[A4B]  353:   IF Parameter$ = "" THEN zaehler = Ø: GOTO
               parameter.ok ELSE zaehler = 1
[E2Ø]  354:   '
[892]  355: m.auswertung:
[RPC]  356:   IF INSTR(Parameter$, ",") = Ø THEN param$
              (zaehler) = Parameter$: GOTO parameter.ok
[HG9]  357:   param$(zaehler) = LEFT$(Parameter$, INSTR
              (Parameter$, ",") - 1)
[DM8]  358:   Parameter$ = MID$(Parameter$, INSTR(Param
              eter$, ",") + 1)
[RF9]  359:   IF zaehler < 8 THEN zaehler = zaehler + 1
              : GOTO m.auswertung
[GFS]  36Ø:   PRINT : PRINT "Die Anzahl der übergebenen
               Parameter ist zu hoch oder die Form der Pa
              rameter    falsch beim Aufruf des Makros: "
              ; MNAME$: GOTO do.fehler
[E2Ø]  361:   '
[YW4]  362:   'Form des Makroaufrufs stimmt
[L82]  363: parameter.ok:
[685]  364:   IF marke$ <> "" THEN PRINT #4, marke$
[M95]  365:   OPEN "makros.wrk" FOR INPUT AS #6
[TØ9]  366:   IF EOF(6) THEN GOTO do.fehler: 'Fehler be
              i einem Do-Aufruf
[E2Ø]  367:   '
[VA2]  368: oeffne.mdatei:
[P73]  369:   LINE INPUT #6, DUMMY$
[8F2]  37Ø:   INPUT #6, werte
[GB3]  371:   LINE INPUT #6, SPNAME$
[TX5]  372:   IF SPNAME$ = MNAME$ THEN GOTO makro.da
[BG2]  373: suche.schluss:
[863]  374:   LINE INPUT #6, zeile$
[SN9]  375:   IF zeile$ <> "Makrodefinition: Ende" THEN
               GOTO suche.schluss
[ZR5]  376:   IF NOT EOF(6) THEN GOTO oeffne.mdatei
[551]  377:   CLOSE #6
[VX2]  378:   fehler = fehler + 1
[ME1]  379:   zeile$ = ""
[ØTF]  38Ø:   PRINT #2, "==> Fehler: Das aufgerufene Ma
              kro "; MNAME$; " wurde noch nicht definiert
              .": GOTO m.schluss
[8E1]  381: makro.da:
[BF6]  382:   IF zaehler <> werte THEN warnung = warnun
              g + 1: PRINT #2, "==> Warnung: Anzahl der P
              arameter bei Definition und Aufruf stimmen
              nicht       überein. Überzählige werden ign
              oriert, Zeilen mit nicht angegebenen Parame
              tern  gelöscht. Makro: ";
[E2Ø]  383:   '
[5Z1]  384: lies.makro:
[863]  385:   LINE INPUT #6, zeile$
[2PH]  386:   IF zeile$ = "Makrodefinition: Ende" THEN
              CLOSE #6: zeile$ = "": GOTO m.schluss: 'End
              e und Zeile nicht abspeichern
[E2Ø]  387:   '
[BX7]  388:   FOR ZAEHLER2 = 1 TO 8: 'Höchstens acht Pa
              rameter
[F76]  389:     'Wenn Parameter nicht vorhanden ist,
[9W5]  39Ø:     ' dann weitermachen; Parameter durch
[6F6]  391:     ' angegebenen Wert ersetzen; falls kein
[3T6]  392:     ' Parameter angegeben, dann Zeile lösch
              en
[UE8]  393:     WHILE INSTR(zeile$, "%" + RIGHT$(STR$(Z
              AEHLER2), 1)) <> Ø
[RLP]  394:       zeile$ = LEFT$(zeile$, INSTR(zeile$,
              "%" + RIGHT$(STR$(ZAEHLER2), 1)) - 1) + par
              am$(ZAEHLER2) + MID$(zeile$, INSTR(zeile$,
              "%" + RIGHT$(STR$(ZAEHLER2), 1)) + 2)
[MW5]  395:       IF param$(ZAEHLER2) = "" THEN zeile$
              = ""
[TRØ]  396:     WEND
[M72]  397:   NEXT ZAEHLER2
[285]  398:   IF zeile$ <> "" THEN PRINT #4, zeile$
[MS2]  399:   GOTO lies.makro
[2T1]  4ØØ: m.schluss:
[285]  4Ø1:   IF zeile$ <> "" THEN PRINT #4, zeile$
[5N2]  4Ø2:   GOSUB aus.param
[TRØ]  4Ø3: WEND
[351]  4Ø4: CLOSE #4
[651]  4Ø5: RETURN
[E2Ø]  4Ø6: '
[LY5]  4Ø7: 'Hauptprogramm, das die benötigten
[C74]  4Ø8: ' Unterprogramme aufruft
[3K2]  4Ø9: Hauptprogramm:
[PNØ]  41Ø: CLS
[6BA]  411: GOSUB flex.init:     'Initialisierung der be
              nötigten Variablen
[4FA]  412: GOSUB lauf.eins:     'Lauf 1: Vorbereitung d
              er Makro-Definitionen
[8L8]  413: GOSUB lauf.zwei:     'Lauf 2: 'INCLUDE'und V
              oranpassung
[8B8]  414: GOSUB symbol:        'Lauf 3: Aufbau der Sym
              boltabelle
[X19]  415: IF KEINENDE THEN CLOSE : IF DOS THEN END EL
              SE GOTO menue
[G8C]  416: GOSUB gen.exe:       'Lauf 4: Erzeugen der E
              XE-Datei und der Assemblerlistings
[LZ7]  417: GOSUB assem.schluss: 'Beenden der Übersetzu
              ng
[WS1]  418: GOTO menue
[E2Ø]  419: '
[5H6]  42Ø: 'Dritter Lauf (Aufbau der Symboltabelle)
[HC1]  421: symbol:
[VB6]  422: CLOSE #1: OPEN "work2.wrk" FOR INPUT AS #1
[GG2]  423: KEINENDE = wahr
[X41]  424: lauf = 1
[XU1]  425: codezeig = Ø
[WV1]  426: zeilnum = Ø
[3E2]  427: WHILE NOT EOF(1)
[SL6]  428:   GOSUB lies.zeile: 'Einlesen einer Zeile
[6U5]  429:   GOSUB analyse:    'Zeile übersetzen
[EA6]  43Ø:   IF befehl$ = "ENDS" THEN GOTO schluss.seg
[E2Ø]  431:   '
[VØ6]  432:   'Eintrag des Symbols in die Tabelle
[T56]  433:   IF marke$ <> "" THEN GOSUB symboleintrag
[D16]  434:   IF befehl$ <> "" THEN GOSUB trans.befehl
[1HB]  435:   IF marke$ <> "" OR befehl$ <> "" THEN IF
              NOT KEINENDE THEN GOTO kein.schluss
[2Ø6]  436:   IF befehl$ = "END" THEN KEINENDE = falsch
[362]  437: schluss.seg:
[5N2]  438:   GOSUB aus.param
[TRØ]  439: WEND
[E2Ø]  44Ø: '
[L56]  441: IF NOT KEINENDE THEN GOTO sym.schluss
[A92]  442: kein.schluss:
[69Y]  443: PRINT #2, : PRINT #2, " Der END-Befehl wurd
              e zu früh oder gar nicht benutzt. SAM III u
              nterbricht die": PRINT #2, "       Übersetz
              ung der SAM-Datei. Drücken Sie bitte eine b
              eliebige Taste !"
[5H3]  444: WHILE INKEY$ = "": WEND
[GG2]  445: KEINENDE = wahr
[R82]  446: sym.schluss:
[651]  447: RETURN
[E2Ø]  448: '
[BV6]  449: 'Vierter Lauf (Erzeugen der EXE-Datei und
[G24]  45Ø: ' des Assemblerlistings)
[3D1]  451: gen.exe:
[LM2]  452: GOSUB init.drei
[E2Ø]  453: '
[3E2]  454: WHILE NOT EOF(1)
[SL6]  455:   GOSUB lies.zeile: 'Einlesen einer Zeile
[YR5]  456:   GOSUB analyse: 'Zeile übersetzen
[GGB]  457:   IF marke$ <> "" THEN GOSUB Phasenfehler:
              'Überprüfen auf Phasenfehler
[D16]  458:   IF befehl$ <> "" THEN GOSUB trans.befehl
[N7A]  459:   GOSUB code.ausgabe: 'Ausgabe des Assemble
              rcodes und -listings
[5N2]  46Ø:   GOSUB aus.param
[M97]  461:   taste$ = INKEY$: IF taste$ = "" THEN GOTO
               weiter
[PR3]  462:   WHILE INKEY$ <> "": WEND
[5H3]  463:   WHILE INKEY$ = "": WEND
[XA1]  464: weiter:
[TRØ]  465: WEND
[E2Ø]  466: '
[651]  467: RETURN
[E2Ø]  468: '
[LV6]  469: 'Unterprogramm zum Einlesen einer Zeile
[GZ1]  47Ø: lies.zeile:
[Y53]  471: LINE INPUT #1, zeile$
```

```
[MC3]  472: zeilnum = zeilnum + 1
[1E3]  473: offsetbedarf = nichts
[AF2]  474: dsflag = falsch
[HG2]  475: befehllaenge = Ø
[E2Ø]  476: '
[651]  477: RETURN
[E2Ø]  478: '
[L86]  479: 'Unterprogramm, in dem die eingelesene
[1X3]  48Ø: ' Zeile übersetzt wird
[ND1]  481: analyse:
[TØ2]  482: zeilzeig1 = 1
[VØ2]  483: zeilzeig2 = 1
[HE1]  484: marke$ = ""
[SK1]  485: befehl$ = ""
[GF1]  486: quell$ = ""
[NA1]  487: ziel$ = ""
[MN4]  488: endzeig = INSTR(zeile$, ";") - 1
[E2Ø]  489: '
[P96]  49Ø: IF endzeig = -1 THEN endzeig = LEN(zeile$)
[KT5]  491: IF endzeig = Ø THEN GOTO ana.schluss
[HE9]  492: GOSUB klein.gross: 'Klein- in Großbuchstabe
           n umwandeln
[E2Ø]  493: '
[AR8]  494: IF INSTR(trenn$, LEFT$(zeile$, 1)) THEN GOT
           O op.analyse
[4Ø2]  495: GOSUB Aufbau
[9Ø2]  496: marke$ = FELD$
[UV9]  497: IF RIGHT$(marke$, 1) = ":" THEN marke$ = LE
           FT$(marke$, LEN(marke$) - 1)
[E2Ø]  498: '
[UT3]  499: 'Analyse des Operanden
[CY1]  5ØØ: op.analyse:
[4Ø2]  5Ø1: GOSUB Aufbau
[L36]  5Ø2: IF NOT gefunden THEN GOTO ana.schluss
[Z72]  5Ø3: befehl$ = FELD$
[AB3]  5Ø4: opndzeig = zeilzeig1
[4Ø2]  5Ø5: GOSUB Aufbau
[L36]  5Ø6: IF NOT gefunden THEN GOTO ana.schluss
[2Y1]  5Ø7: ziel$ = FELD$
[4Ø2]  5Ø8: GOSUB Aufbau
[L36]  5Ø9: IF NOT gefunden THEN GOTO ana.schluss
[812]  51Ø: quell$ = FELD$
[252]  511: ana.schluss:
[E2Ø]  512: '
[651]  513: RETURN
[E2Ø]  514: '
[3C7]  515: 'Unterprogramm, das die in der eingelesenen
[3P6]  516: ' Zeile vorhandenen Kleinbuchstaben in
[1W4]  517: ' Großbuchstaben umwandelt
[842]  518: klein.gross:
[163]  519: FOR i = 1 TO endzeig
[P63]  52Ø:   b$ = MID$(zeile$, i, 1)
[9VF]  521:   IF b$ = "'" THEN ZEICHENDE = INSTR(i + 1,
           zeile$, b$): IF ZEICHENDE > Ø THEN i = ZEI
           CHENDE: GOTO kommentar
[AAB]  522:   IF ASC(b$) >= 97 AND ASC(b$) <= 122 THEN
           b$ = CHR$(ASC(b$) - 32): MID$(zeile$, i, 1)
           = b$
[VU1]  523: kommentar:
[U11]  524: NEXT i
[E2Ø]  525: '
[651]  526: RETURN
[E2Ø]  527: '
[YD6]  528: 'Unterprogramm, das den Befehlsaufbau
[UØ2]  529: ' analysiert
[4A1]  53Ø: Aufbau:
[BB4]  531: WHILE zeilzeig1 <= endzeig
[H4A]  532:   IF INSTR(trenn$, MID$(zeile$, zeilzeig1,
           1)) = Ø THEN GOTO marke.1
[8W3]  533:   zeilzeig1 = zeilzeig1 + 1
[TRØ]  534: WEND
[E2Ø]  535: '
[281]  536: marke.1:
[BM6]  537: IF zeilzeig1 <= endzeig THEN GOTO marke.2
[PP2]  538: gefunden = falsch
[651]  539: RETURN
[E2Ø]  54Ø: '
[681]  541: marke.2:
[4S7]  542: IF MID$(zeile$, zeilzeig1, 1) <> "'" THEN G
           OTO marke.3
[ØU6]  543: ZEICHENDE = INSTR(zeilzeig1 + 1, zeile$, "'
           ")
[F55]  544: IF ZEICHENDE = Ø THEN GOTO marke.3
[PW3]  545: zeilzeig2 = ZEICHENDE + 1
[4A3]  546: GOTO feldzuweisung
[E2Ø]  547: '
[A81]  548: marke.3:
[WF3]  549: zeilzeig2 = zeilzeig1
[FB4]  55Ø: WHILE zeilzeig2 <= endzeig
[1EB]  551:   IF INSTR(trenn$, MID$(zeile$, zeilzeig2,
           1)) > Ø THEN GOTO feldzuweisung
[CW3]  552:   zeilzeig2 = zeilzeig2 + 1
[TRØ]  553: WEND
[E2Ø]  554: '
[CM2]  555: feldzuweisung:
[8F8]  556: FELD$ = MID$(zeile$, zeilzeig1, zeilzeig2 -
           zeilzeig1)
[WF3]  557: zeilzeig1 = zeilzeig2
[4G2]  558: gefunden = wahr
[651]  559: RETURN
[E2Ø]  56Ø: '
[7Ø7]  561: 'Unterprogramm, das ein bisher noch nicht
[MN7]  562: ' definiertes Symbol in die Tabelle einträg
           t
[3L2]  563: symboleintrag:
[ZG2]  564: objekt$ = marke$
[A73]  565: GOSUB suche.symbol
[7N6]  566: IF NOT gefunden THEN GOTO sym.ueberlauf
[VX2]  567: fehler = fehler + 1
[2GD]  568: PRINT #2, "==> Fehler: Label "; marke$; " i
           n "; zeilnum; "zum wiederholten Mal definie
           rt."
[651]  569: RETURN
[E2Ø]  57Ø: '
[MF2]  571: sym.ueberlauf:
[6V6]  572: IF aktsymbol < 5ØØ THEN GOTO sym.eintragen
[VX2]  573: fehler = fehler + 1
[58C]  574: PRINT #2, "==> Fehler: Zu viele benutzerdef
           inierte Symbole in Zeile"; zeilnum
[651]  575: RETURN
[E2Ø]  576: '
[PG2]  577: sym.eintragen:
[YØ4]  578: aktsymbol = aktsymbol + 1
[TC4]  579: symbol$(aktsymbol) = marke$
[HL4]  58Ø: wert1(aktsymbol) = codezeig
[MF4]  581: symboltyp(aktsymbol) = nah
[651]  582: RETURN
[E2Ø]  583: '
[Y27]  584: 'Unterprogramm, das die Symboltabelle auf
[E64]  585: ' Phasenfehler untersucht
[E72]  586: Phasenfehler:
[LP9]  587: IF befehl$ = "EQU" OR befehl$ = "ENDS" THEN
           GOTO phasen.schluss
[ZG2]  588: objekt$ = marke$
[A73]  589: GOSUB suche.symbol
[K9D]  59Ø: IF (symboltyp(tabzeiger) AND (nah OR speich
           er)) = falsch THEN GOTO phasen.schluss
[VB9]  591: IF wert1(tabzeiger) = codezeig THEN GOTO ph
           asen.schluss
[E2Ø]  592: '
[HP8]  593: fehler = fehler + 1: PRINT #2, "==> Fehler:
           Phasenfehler"
[SN2]  594: phasen.schluss:
[E2Ø]  595: '
[651]  596: RETURN
[E2Ø]  597: '
[347]  598: 'Unterprogramm, das die angegebene Adresse
[UH4]  599: ' in der Symboltabelle sucht
[E82]  6ØØ: suche.symbol:
[TØ5]  6Ø1: FOR tabzeiger = 1 TO aktsymbol
[83B]  6Ø2:   IF symbol$(tabzeiger) = objekt$ THEN GOTO
           symbol.da: 'Symbol gefunden
[FG2]  6Ø3: NEXT tabzeiger
[PP2]  6Ø4: gefunden = falsch
[651]  6Ø5: RETURN
[E2Ø]  6Ø6: '
[NP1]  6Ø7: symbol.da:
[4G2]  6Ø8: gefunden = wahr
[651]  6Ø9: RETURN
[E2Ø]  61Ø: '
[S56]  611: 'Unterprogramm, das den angegebenen
[VN6]  612: ' Assemblerbefehl in der Kommandotabelle
[X71]  613: ' sucht
[F82]  614: befehlssuche:
[6E1]  615: MOVE = 231
[RD2]  616: START = MOVE / 2
[E2Ø]  617: '
[Y92]  618: WHILE MOVE >= 2
[2A2]  619:   MOVE = MOVE / 2
[KLC]  62Ø:   IF befehl$ > kommando$(START) THEN START
           = START + MOVE ELSE START = START - MOVE
[654]  621:   IF START < 1 THEN START = 1
[3K4]  622:   IF START > 231 THEN START = 231
[TRØ]  623: WEND
[E2Ø]  624: '
[DF4]  625: FOR operzeig = START TO 231
[AA8]  626:   IF kommando$(operzeig) > befehl$ THEN GOT
           O da.falsch
[Ø69]  627:   IF kommando$(operzeig) <> befehl$ THEN GO
           TO such.schluss
[3ZB]  628:   IF (quellregtyp(operzeig) AND quelltyp) =
           falsch THEN GOTO such.schluss
[XLB]  629:   IF (zielregtyp(operzeig) AND zieltyp) = f
           alsch THEN GOTO such.schluss
[V12]  63Ø:   GOTO da.wahr
[CA2]  631: such.schluss:
[EB2]  632: NEXT operzeig
[E2Ø]  633: '
[WL1]  634: da.falsch:
[PP2]  635: gefunden = falsch
[651]  636: RETURN
[E2Ø]  637: '
[P91]  638: da.wahr:
[4G2]  639: gefunden = wahr
[651]  64Ø: RETURN
[E2Ø]  641: '
```

Listing 1. »sam3.asc« ist die neue Version 1.1 von SAM III

```
[8V5]  642: 'Unterprogramm, das die angegebene
[GS3]  643: ' Operation übersetzt
[852]  644: trans.befehl:
[S5N]  645: IF befehl$ = "TITLE" OR befehl$ = "SUBTTL"
            OR befehl$ = "EVEN" OR befehl$ = "ENDS" OR
            befehl$ = "SEGMENT" OR befehl$ = "END" THEN
             GOTO bef.suche
[E2Ø]  646: '
[WC2]  647: objekt$ = ziel$
[BV2]  648: GOSUB op.auswert
[YW2]  649: zieltyp = objtyp
[2E3]  65Ø: zielwert1 = objwert1
[5E3]  651: zielwert2 = objwert2
[E2Ø]  652: '
[78C]  653: IF befehl$ = "RET" THEN quelltyp = prozedur
            typ(stapelzeiger): GOTO bef.suche
[ZN5]  654: objekt$ = quell$: GOSUB op.auswert
[UX2]  655: quelltyp = objtyp
[PF3]  656: quellwert1 = objwert1
[VF3]  657: quellwert2 = objwert2
[E2Ø]  658: '
[V36]  659: 'Suche Kommando in der Befehlstabelle
[AP1]  66Ø: bef.suche:
[9K2]  661: objekt$ = befehl$
[953]  662: GOSUB befehlssuche
[U26]  663: IF gefunden THEN GOTO flag.zuweisung
[E2Ø]  664: '
[ZR3]  665: IF lauf = 1 THEN RETURN
[HEA]  666: fehler = fehler + 1: PRINT #2, "==> Fehler:
             Syntax-Fehler: "; befehl$
[E2Ø]  667: '
[35G]  668: IF ((akku8 OR akku16 OR reg8 OR reg16 OR se
            greg OR cs) AND (zieltyp OR quelltyp)) THEN
             GOTO s.schluss
[1SC]  669: IF (quelltyp AND (nichts OR direkt8 OR dire
            kt16)) = falsch THEN GOTO s.schluss
[DU8]  67Ø: IF INSTR("BW", RIGHT$(befehl$, 1)) <> Ø THE
            N GOTO s.schluss
[E2Ø]  671: '
[UC3]  672: warnung = warnung + 1
[5FC]  673: PRINT #2, "==> Warnung: Achten Sie auf Wort
            - beziehungsweise Byte-Operationen"
[ET1]  674: s.schluss:
[E2Ø]  675: '
[651]  676: RETURN
[E2Ø]  677: '
[FT2]  678: flag.zuweisung:
[BA3]  679: flag = flag(operzeig)
[1ØA]  68Ø: IF flag AND operand THEN GOSUB gen.code ELS
            E GOSUB pseudo.ops
[651]  681: RETURN
[E2Ø]  682: '
[FR5]  683: 'Unterprogramm, das den Operanden
[ZY1]  684: ' auswertet
[L22]  685: op.auswert:
[WØ6]  686: IF LEN(objekt$) > Ø THEN GOTO obj.suche
[8N2]  687: objtyp = nichts
[651]  688: RETURN
[E2Ø]  689: '
[BR1]  69Ø: obj.suche:
[A73]  691: GOSUB suche.symbol
[KN5]  692: IF NOT gefunden THEN GOTO w.objekt
[8Y4]  693: objtyp = symboltyp(tabzeiger)
[RK4]  694: objwert1 = wert1(tabzeiger)
[N78]  695: IF tabzeiger <= 29 THEN objwert2 = wert2(ta
            bzeiger)
[651]  696: RETURN
[E2Ø]  697: '
[DG1]  698: w.objekt:
[U62]  699: GOSUB op.zahl
[E75]  7ØØ: IF NOT gefunden THEN GOTO w2.obj
[1T2]  7Ø1: objtyp = zahltyp
[6B3]  7Ø2: objwert1 = zahlwert
[651]  7Ø3: RETURN
[E2Ø]  7Ø4: '
[N41]  7Ø5: w2.obj:
[C83]  7Ø6: GOSUB speichertest
[F75]  7Ø7: IF NOT gefunden THEN GOTO w3.obj
[1V2]  7Ø8: objtyp = speicher
[AR3]  7Ø9: objwert1 = SPEICHERADR
[651]  71Ø: RETURN
[E2Ø]  711: '
[S41]  712: w3.obj:
[SR2]  713: GOSUB op.offset
[G75]  714: IF NOT gefunden THEN GOTO w4.obj
[KD3]  715: objtyp = speicherreg
[D23]  716: objwert1 = regwert
[651]  717: RETURN
[E2Ø]  718: '
[V41]  719: w4.obj:
[6W2]  72Ø: GOSUB opisoffset
[H75]  721: IF NOT gefunden THEN GOTO w5.obj
[8W2]  722: objtyp = OFFSETYP
[MK3]  723: objwert1 = OFFSETWERT
[651]  724: RETURN
[E2Ø]  725: '
[Y41]  726: w5.obj:
[ZS2]  727: GOSUB zeichen.op
[K75]  728: IF NOT gefunden THEN GOTO w6.obj
[5P4]  729: objtyp = direkt8 OR direkt16
[7C3]  73Ø: objwert1 = zeichwert
[651]  731: RETURN
[E2Ø]  732: '
[151]  733: w6.obj:
[587]  734: IF LEFT$(objekt$, 1) <> "'" THEN GOTO sym.f
            ehler
[AN2]  735: objtyp = zeichen
[651]  736: RETURN
[E2Ø]  737: '
[222]  738: sym.fehler:
[ØRE]  739: IF lauf = 2 THEN PRINT #2, "==> Fehler: Sym
            bol "; objekt$; " nicht definiert": fehler
            = fehler + 1
[1Ø4]  74Ø: objtyp = nah OR speicher
[E2Ø]  741: '
[651]  742: RETURN
[E2Ø]  743: '
[BL6]  744: 'Unterprogramm, das überprüft, ob der
[VS3]  745: ' Operand eine Zahl ist
[4B1]  746: op.zahl:
[PP2]  747: gefunden = falsch
[5Y1]  748: tn$ = objekt$
[WV5]  749: IF RIGHT$(tn$, 1) <> "H" THEN GOTO binaer
[E2Ø]  75Ø: '
[G54]  751: tn$ = LEFT$(tn$, LEN(tn$) - 1)
[8LØ]  752: i = 1
[E2Ø]  753: '
[863]  754: FOR i = 1 TO LEN(tn$)
[ME2]  755:   b$ = MID$(tn$, i, 1)
[CX6]  756:   IF INSTR("Ø123456789ABCDEF", b$) = Ø THEN
             RETURN
[U11]  757: NEXT i
[E2Ø]  758: '
[MP3]  759: zahlwert = VAL("&H" + tn$)
[X22]  76Ø: GOTO zahl.ok
[E2Ø]  761: '
[ØA1]  762: binaer:
[1L6]  763: IF RIGHT$(tn$, 1) <> "B" THEN GOTO zahl.tra
            ns
[G54]  764: tn$ = LEFT$(tn$, LEN(tn$) - 1)
[8LØ]  765: i = 1
[E2Ø]  766: '
[863]  767: FOR i = 1 TO LEN(tn$)
[ME2]  768:   b$ = MID$(tn$, i, 1)
[PS4]  769:   IF INSTR("Ø1", b$) = Ø THEN RETURN
[U11]  77Ø: NEXT i
[E2Ø]  771: '
[3X1]  772: zahlwert = Ø
[E2Ø]  773: '
[WB4]  774: FOR i = LEN(tn$) TO 1 STEP -1
[UT8]  775:   zahlwert = zahlwert + VAL(MID$(tn$, i, 1)
            ) * 2 ^ (LEN(tn$) - i)
[U11]  776: NEXT i
[E2Ø]  777: '
[X22]  778: GOTO zahl.ok
[122]  779: zahl.trans:
[E2Ø]  78Ø: '
[863]  781: FOR i = 1 TO LEN(tn$)
[ME2]  782:   b$ = MID$(tn$, i, 1)
[L46]  783:   IF INSTR("Ø123456789-+", b$) = Ø THEN RET
            URN
[U11]  784: NEXT i
[E2Ø]  785: '
[FY2]  786: zahlwert = VAL(tn$)
[KC1]  787: zahl.ok:
[4G2]  788: gefunden = wahr
[M4D]  789: IF LEN(HEX$(zahlwert)) < 3 THEN zahltyp = d
            irekt16 OR direkt8 ELSE zahltyp = direkt16
[E2Ø]  79Ø: '
[651]  791: RETURN
[E2Ø]  792: '
[2M5]  793: 'Unterprogramm, das den direkten
[BU4]  794: ' Speicherzugriff auswertet
[2A2]  795: speichertest:
[3Y1]  796: MR$ = objekt$
[SP8]  797: IF LEFT$(MR$, 1) <> "[" OR RIGHT$(MR$, 1) <
            > "]" THEN RETURN
[E2Ø]  798: '
[ZN4]  799: objekt$ = MID$(MR$, 2, LEN(MR$) - 2)
[U62]  8ØØ: GOSUB op.zahl
[5H9]  8Ø1: IF gefunden THEN SPEICHERADR = zahlwert: GO
            TO ok.speicher
[A73]  8Ø2: GOSUB suche.symbol
[1U6]  8Ø3: IF NOT gefunden THEN GOTO falsch.speicher
[73C]  8Ø4: IF (symboltyp(tabzeiger) AND direkt16) = fa
            lsch THEN GOTO falsch.speicher
[615]  8Ø5: SPEICHERADR = wert1(tabzeiger)
[SU2]  8Ø6: GOTO ok.speicher
[E2Ø]  8Ø7: '
[KR2]  8Ø8: falsch.speicher:
[PP2]  8Ø9: gefunden = falsch
[VX1]  81Ø: objekt$ = MR$
[651]  811: RETURN
[E2Ø]  812: '
[122]  813: ok.speicher:
[VX1]  814: objekt$ = MR$
[651]  815: RETURN
[E2Ø]  816: '
```

```
[5A6]  817: 'Unterprogramm, das den Operanden als
[216]  818: ' Offset zu einem Register behandelt
[UP1]  819: op.offset:
[ZM2]  82Ø: pdoder$ = objekt$
[KPD]  821: IF objekt$ = "[BP]" THEN regwert = 6: offse
            tbedarf = direkt8: offset = Ø: GOTO ein.sym
            bol
[984]  822: zeiger = INSTR(objekt$, "[")
[6T5]  823: IF zeiger <= 1 THEN GOTO null.symbol
[KN7]  824: regis$ = RIGHT$(pdoder$, LEN(pdoder$) - zei
            ger + 1)
[GK7]  825: IF regis$ = "[BP]" THEN regwert = 6: GOTO o
            p.disp
[LK2]  826: objekt$ = regis$
[A73]  827: GOSUB suche.symbol
[E2Ø]  828: '
[MRC]  829: IF NOT gefunden OR symboltyp(tabzeiger) <>
            speicherreg THEN GOTO null.symbol
[FC4]  83Ø: regwert = wert1(tabzeiger)
[FC1]  831: op.disp:
[3Y4]  832: disp$ = LEFT$(pdoder$, zeiger - 1)
[CB2]  833: objekt$ = disp$
[A73]  834: GOSUB suche.symbol
[E2Ø]  835: '
[526]  836: IF NOT gefunden THEN GOTO kein.symbol
[D8D]  837: IF (symboltyp(tabzeiger) AND (direkt16 OR d
            irekt8)) = falsch THEN GOTO kein.symbol
[D16]  838: offsetbedarf = symboltyp(tabzeiger)
[CØ4]  839: offset = wert1(tabzeiger)
[2U2]  84Ø: GOTO ein.symbol
[E2Ø]  841: '
[S52]  842: kein.symbol:
[U62]  843: GOSUB op.zahl
[UZC]  844: IF gefunden THEN offsetbedarf = zahltyp: of
            fset = zahlwert: GOTO ein.symbol
[6W2]  845: GOSUB opisoffset
[136]  846: IF NOT gefunden THEN GOTO null.symbol
[F14]  847: offsetbedarf = OFFSETYP
[NC3]  848: offset = OFFSETWERT
[E2Ø]  849: '
[W22]  85Ø: ein.symbol:
[9M2]  851: objekt$ = pdoder$
[4G2]  852: gefunden = wahr
[E2Ø]  853: '
[651]  854: RETURN
[E2Ø]  855: '
[S72]  856: null.symbol:
[9M2]  857: objekt$ = pdoder$
[PP2]  858: gefunden = falsch
[651]  859: RETURN
[E2Ø]  86Ø: '
[5A6]  861: 'Unterprogramm, das den Operanden als
[4A3]  862: ' Offset deklariert
[E22]  863: opisoffset:
[AY1]  864: os$ = objekt$
[CE9]  865: IF LEFT$(os$, 7) <> "OFFSET(" THEN gefunden
             = falsch: RETURN
[194]  866: IF lauf = 1 THEN GOTO arg.ok
[H36]  867: objekt$ = MID$(objekt$, 8, LEN(objekt$) - 8
            )
[A73]  868: GOSUB suche.symbol
[E2Ø]  869: '
[AHH]  87Ø: IF gefunden AND (symboltyp(tabzeiger) AND (
            speicher OR nah)) THEN OFFSETWERT = wert1(t
            abzeiger): GOTO arg.ok
[VX2]  871: fehler = fehler + 1
[S9B]  872: PRINT #2, "==> Fehler: Unerlaubtes oder und
            efiniertes Offset-Argument"
[F92]  873: OFFSETWERT = Ø
[E2Ø]  874: '
[991]  875: arg.ok:
[4G2]  876: gefunden = wahr
[A63]  877: OFFSETYP = direkt16
[4Y1]  878: objekt$ = os$
[651]  879: RETURN
[E2Ø]  88Ø: '
[5A6]  881: 'Unterprogramm, das den Operanden als
[493]  882: ' Zeichen definiert
[512]  883: zeichen.op:
[PP2]  884: gefunden = falsch
[SV4]  885: IF LEN(objekt$) <> 3 THEN RETURN
[SY5]  886: IF LEFT$(objekt$, 1) <> "'" THEN RETURN
[UU5]  887: IF RIGHT$(objekt$, 1) <> "'" THEN RETURN
[E2Ø]  888: '
[4G2]  889: gefunden = wahr
[LG5]  89Ø: zeichwert = ASC(MID$(objekt$, 2, 1))
[651]  891: RETURN
[E2Ø]  892: '
[ZN6]  893: 'Unterprogramm, das im zweiten Lauf den
[TP3]  894: ' Programmcode erzeugt
[ZF1]  895: gen.code:
[7S2]  896: GOSUB wort.byte
[3L3]  897: codezeig = codezeig + 1
[CB5]  898: IF lauf = 2 THEN GOSUB code.aufbau
[G5C]  899: IF (operwert(operzeig) <> 213) AND (operwer
            t(operzeig) <> 212) THEN GOTO gen.m1
[E2Ø]  9ØØ: '
[3L3]  9Ø1: codezeig = codezeig + 1
```

```
[UØC]  9Ø2: IF lauf = 2 THEN befehllaenge = befehllaeng
            e + 1: befehltab(befehllaenge) = 1Ø
[E2Ø]  9Ø3: '
[S61]  9Ø4: gen.m1:
[NH6]  9Ø5: IF offsetbedarf = nichts THEN GOTO gen.m2
[GMD]  9Ø6: IF (offsetbedarf AND direkt8) THEN codezeig
             = codezeig + 1 ELSE codezeig = codezeig +
            2
[E2Ø]  9Ø7: '
[V61]  9Ø8: gen.m2:
[64B]  9Ø9: IF (flag AND (modbytebedarf OR erweibedarf)
            ) = falsch THEN GOTO gen.m3
[ERA]  91Ø: IF (zieltyp OR quelltyp) AND speicher THEN
            codezeig = codezeig + 2
[E2Ø]  911: '
[Y61]  912: gen.m3:
[458]  913: IF (flag AND erweibedarf) = falsch THEN GOT
            O gen.m4
[3L3]  914: codezeig = codezeig + 1
[BY4]  915: IF lauf = 2 THEN GOSUB erw.byte
[E2Ø]  916: '
[171]  917: gen.m4:
[EA8]  918: IF (flag AND modbytebedarf) = falsch THEN G
            OTO gen.m5
[3L3]  919: codezeig = codezeig + 1
[CA8]  92Ø: GOSUB adr.art: codezeig = codezeig + offzus
            atz '<- ***
[E2Ø]  921: '
[471]  922: gen.m5:
[438]  923: IF (flag AND disp8bedarf) = falsch THEN GOT
            O gen.m6
[3L3]  924: codezeig = codezeig + 1
[EP5]  925: IF lauf = 2 THEN GOSUB diff.relativ
[E2Ø]  926: '
[771]  927: gen.m6:
[EØ8]  928: IF (flag AND disp16bedarf) = falsch THEN GO
            TO gen.m7
[7L3]  929: codezeig = codezeig + 2
[A74]  93Ø: IF lauf = 2 THEN GOSUB dis16
[E2Ø]  931: '
[A71]  932: gen.m7:
[Y78]  933: IF (flag AND direkt8bedarf) = falsch THEN G
            OTO gen.m8
[3L3]  934: codezeig = codezeig + 1
[PN4]  935: IF lauf = 2 THEN GOSUB byte.di
[E2Ø]  936: '
[D71]  937: gen.m8:
[ZY9]  938: IF wort OR ((flag AND direkt16bedarf) = fal
            sch) THEN GOTO gen.m9
[3L3]  939: codezeig = codezeig + 1
[PN4]  94Ø: IF lauf = 2 THEN GOSUB byte.di
[E2Ø]  941: '
[G71]  942: gen.m9:
[T2B]  943: IF NOT (wort) OR ((flag AND direkt16bedarf)
             = falsch) THEN GOTO gen.m1Ø
[W8C]  944: IF zieltyp = direkt16 THEN codezeig = codez
            eig + 4 ELSE codezeig = codezeig + 2
[DR4]  945: IF lauf = 2 THEN GOSUB wort.di
[E2Ø]  946: '
[A71]  947: gen.m1Ø:
[RT9]  948: IF (flag AND speicherbedarf) = falsch THEN
            GOTO gen.schluss
[7L3]  949: codezeig = codezeig + 2
[Z25]  95Ø: IF lauf = 2 THEN GOSUB baut.16bit
[E2Ø]  951: '
[762]  952: gen.schluss:
[651]  953: RETURN
[E2Ø]  954: '
[HW6]  955: 'Unterprogramme, das zwischen Wort- und
[K35]  956: ' Byte-Operanden unterscheidet
[9U1]  957: wort.byte:
[EWC]  958: IF operwert(operzeig) = 246 THEN IF quellty
            p AND reg8 THEN GOTO wort.nein '<- ***
[UYC]  959: IF (zieltyp OR quelltyp) AND (reg16 OR akku
            16 OR segreg OR cs) THEN GOTO wort.ja
[1DA]  96Ø: IF (zieltyp OR quelltyp) AND (reg8 OR akku8
            ) THEN GOTO wort.nein
[D27]  961: IF RIGHT$(befehl$, 1) = "B" THEN GOTO wort.
            nein
[FD1]  962: wort.ja:
[7X1]  963: wort = wahr
[651]  964: RETURN
[E2Ø]  965: '
[4T1]  966: wort.nein:
[T32]  967: wort = falsch
[651]  968: RETURN
[E2Ø]  969: '
[VC6]  97Ø: 'Unterprogramm, das die Pseudobefehle
[8W1]  971: ' behandelt
[ZZ1]  972: pseudo.ops:
[V5L]  973: ON operwert(operzeig) GOSUB ps.equ, ps.org,
             ps.db, ps.ds, ps.proc, ps.endp, ps.title,
            ps.even, ps.end, ps.segment, ps.ends
[XLD]  974: '                          EQU      ORG
             DB      DS      PROC      ENDP      TITLE
           EVEN      END      SEGMENT      ENDS
[651]  975: RETURN
```

Listing 1. »sam3.asc« ist die neue Version 1.1 von SAM III

```
[E2Ø]  976: '
[2Z5]  977: 'Pseudobefehl 'END' (erfüllt keinen
[673]  978: ' besonderen Zweck)
[671]  979: ps.end:
[651]  98Ø: RETURN
[E2Ø]  981: '
[E93]  982: 'Pseudobefehl 'EVEN'
[G52]  983: ps.even: '*** ->
[LK8]  984: IF codezeig / 2 = INT(codezeig / 2) THEN GO
           TO ps.noeven
[6GD]  985: codezeig = codezeig + 1: befehllaenge = bef
           ehllaenge + 1: befehltab(befehllaenge) = 14
           4
[LR1]  986: ps.noeven:
[8Y1]  987: RETURN '<- ***
[LR7]  988: 'Unterprogramm, das den Programmcode aufbau
           t
[P22]  989: code.aufbau:
[ZX4]  99Ø: befehllaenge = befehllaenge + 1
[MB7]  991: befehltab(befehllaenge) = operwert(operzeig
           )
[E2Ø]  992: '
[9S7]  993: IF (flag AND regsum) = falsch THEN GOTO auf
           bau.m2
[8U9]  994: IF zieltyp AND (segreg OR cs) THEN r = ziel
           wert2: GOTO aufbau.m1
[GKA]  995: IF (flag AND richtung) THEN r = quellwert2
           / 8 ELSE r = zielwert2 / 8
[E2Ø]  996: '
[HL1]  997: aufbau.m1:
[SD8]  998: befehltab(befehllaenge) = befehltab(befehll
           aenge) + r
[E2Ø]  999: '
[LL1] 1ØØØ: aufbau.m2:
[PN7] 1ØØ1: IF (flag AND konst2) = falsch THEN GOTO auf
           bau.m3
[8DA] 1ØØ2: IF wort THEN befehltab(befehllaenge) = befe
           hltab(befehllaenge) + 1
[E2Ø] 1ØØ3: '
[NL1] 1ØØ4: aufbau.m3:
[MP8] 1ØØ5: IF (flag AND konst1) = falsch THEN GOTO auf
           bau.schluss
[6AC] 1ØØ6: IF quelltyp AND cl THEN befehltab(befehllae
           nge) = befehltab(befehllaenge) + 2
[E2Ø] 1ØØ7: '
[HP2] 1ØØ8: aufbau.schluss:
[651] 1ØØ9: RETURN
[E2Ø] 1Ø1Ø: '
[277] 1Ø11: 'Unterprogramm, das das Erweiterungsbyte
[B61] 1Ø12: ' bildet
[2N1] 1Ø13: erw.byte:
[YF1] 1Ø14: maske = 56
[T33] 1Ø15: ext = flag AND maske
[RFA] 1Ø16: IF flag AND richtung THEN zielwert2 = ext E
           LSE quellwert2 = ext
[M32] 1Ø17: GOSUB adr.art
[E2Ø] 1Ø18: '
[651] 1Ø19: RETURN
[E2Ø] 1Ø2Ø: '
[G47] 1Ø21: 'Unterprogramm, das die Adressierungsart
[8W1] 1Ø22: ' behandelt
[6E1] 1Ø23: adr.art:
[918] 1Ø24: befehllaenge = befehllaenge + 1: offzusatz
           = Ø '<- ***
[Z8B] 1Ø25: IF ((zieltyp OR quelltyp) AND speicher) = f
           alsch THEN GOTO adr.weiter
[ME9] 1Ø26: IF zieltyp = speicher THEN m = quellwert2 E
           LSE m = zielwert2
[ZU4] 1Ø27: befehltab(befehllaenge) = 6 + m
[2M2] 1Ø28: GOSUB baut.16bit
[E2Ø] 1Ø29: '
[651] 1Ø3Ø: RETURN
[E2Ø] 1Ø31: '
[112] 1Ø32: adr.weiter:
[9VD] 1Ø33: IF flag AND richtung THEN m = quellwert1 +
           zielwert2 ELSE m = zielwert1 + quellwert2
[HD4] 1Ø34: befehltab(befehllaenge) = m
[E2Ø] 1Ø35: '
[XK7] 1Ø36: IF offsetbedarf = nichts THEN GOTO adr.schl
           uss
[W67] 1Ø37: IF offset > 127 OR offset < -128 THEN GOTO
           adr.w2
[WG8] 1Ø38: befehltab(befehllaenge) = befehltab(befehll
           aenge) + 64
[1K9] 1Ø39: IF offset < Ø THEN offset = offset AND 255:
            offzusatz = -1 '<- ***
[ZX4] 1Ø4Ø: befehllaenge = befehllaenge + 1
[C75] 1Ø41: befehltab(befehllaenge) = offset
[E2Ø] 1Ø42: '
[651] 1Ø43: RETURN
[E2Ø] 1Ø44: '
[G71] 1Ø45: adr.w2:
[CR8] 1Ø46: befehltab(befehllaenge) = befehltab(befehll
           aenge) + 128
[3Y4] 1Ø47: befehllaenge = befehllaenge + 2
[CW4] 1Ø48: lsb = offset: GOSUB teil.16bit
[4Ø5] 1Ø49: befehltab(befehllaenge - 1) = lsb
[X1A] 1Ø5Ø: befehltab(befehllaenge) = msb: IF msb = Ø T
           HEN offzusatz = 1 '<- ***
```

```
[862] 1Ø51: adr.schluss:
[E2Ø] 1Ø52: '
[651] 1Ø53: RETURN
[E2Ø] 1Ø54: '
[AB7] 1Ø55: 'Unterprogramm, das eine 16-Bit-Zahl in den
[TK7] 1Ø56: ' niedrigwertigen und höherwertigen Bereich
[C22] 1Ø57: ' unterteilt
[6V1] 1Ø58: teil.16bit:
[E2Ø] 1Ø59: '
[ZV1] 1Ø6Ø: h$ = HEX$(lsb)
[KE4] 1Ø61: h$ = STRING$(4 - LEN(h$), "Ø") + h$
[Z54] 1Ø62: lsb = VAL("&H" + RIGHT$(h$, 2))
[KU3] 1Ø63: msb = VAL("&H" + LEFT$(h$, 2))
[E2Ø] 1Ø64: '
[651] 1Ø65: RETURN
[E2Ø] 1Ø66: '
[PY6] 1Ø67: 'Unterprogramm, das die Differenz für die
[Ø35] 1Ø68: ' relative Adressierung bildet
[Z62] 1Ø69: diff.relativ:
[GU3] 1Ø7Ø: d = zielwert1 - codezeig
[XU4] 1Ø71: IF ABS(d) < 128 THEN GOTO dist.ok
[WKØ] 1Ø72: d = Ø
[E2Ø] 1Ø73: '
[FAG] 1Ø74: IF lauf = 2 THEN PRINT #2, "==> Fehler: Dis
           tanz zu groß für einen 'short jump'": fehle
           r = fehler + 1
[KD1] 1Ø75: dist.ok:
[E2Ø] 1Ø76: '
[PS3] 1Ø77: IF d < Ø THEN d = d AND 255
[ZX4] 1Ø78: befehllaenge = befehllaenge + 1
[DC4] 1Ø79: befehltab(befehllaenge) = d
[E2Ø] 1Ø8Ø: '
[651] 1Ø81: RETURN
[E2Ø] 1Ø82: '
[WW5] 1Ø83: 'Routine, die das 16-Bit-Displacement
[PW1] 1Ø84: ' errechnet
[UZØ] 1Ø85: dis16:
[GU3] 1Ø86: d = zielwert1 - codezeig
[S2H] 1Ø87: IF befehl$ = "JMP" AND d <= 128 THEN PRINT
           #2, "==> Warnung: Sie könnten 'JMPS' benutz
           en": warnung = warnung + 1
[Ø37] 1Ø88: IF d >= Ø OR befehl$ = "CALL" THEN GOTO rue
           ck16
[WKØ] 1Ø89: d = Ø
[E2Ø] 1Ø9Ø: '
[4BH] 1Ø91: IF lauf = 2 THEN PRINT #2, "==> Fehler: Nic
           ht erlaubter Rückwärtssprung ('long jump')"
           : fehler = fehler + 1
[N91] 1Ø92: rueck16:
[E2Ø] 1Ø93: '
[5W3] 1Ø94: lsb = d: GOSUB teil.16bit
[3Y4] 1Ø95: befehllaenge = befehllaenge + 2
[4Ø5] 1Ø96: befehltab(befehllaenge - 1) = lsb
[8M4] 1Ø97: befehltab(befehllaenge) = msb
[E2Ø] 1Ø98: '
[651] 1Ø99: RETURN
[E2Ø] 11ØØ: '
[2M5] 11Ø1: 'Unterprogramm, das den direkten
[2T3] 11Ø2: ' Wortzugriff beachtet
[XD1] 11Ø3: wort.di:
[WPB] 11Ø4: IF zieltyp AND direkt16 THEN direktwert = z
           ielwert1: GOSUB wort.weiter
[B6C] 11Ø5: IF quelltyp AND direkt16 THEN direktwert =
           quellwert1: GOSUB wort.weiter
[E2Ø] 11Ø6: '
[651] 11Ø7: RETURN
[E2Ø] 11Ø8: '
[E72] 11Ø9: wort.weiter:
[2M5] 111Ø: lsb = direktwert: GOSUB teil.16bit
[3Y4] 1111: befehllaenge = befehllaenge + 2
[4Ø5] 1112: befehltab(befehllaenge - 1) = lsb
[8M4] 1113: befehltab(befehllaenge) = msb
[E2Ø] 1114: '
[651] 1115: RETURN
[E2Ø] 1116: '
[2M5] 1117: 'Unterprogramm, das den direkten
[FN3] 1118: ' Bytezugriff beachtet
[AD1] 1119: byte.di:
[PBB] 112Ø: IF zieltyp AND direkt8 THEN direktwert = zi
           elwert1: GOSUB byte.weiter
[REC] 1121: IF quelltyp AND direkt8 THEN direktwert = q
           uellwert1: GOSUB byte.weiter
[E2Ø] 1122: '
[651] 1123: RETURN
[E2Ø] 1124: '
[V62] 1125: byte.weiter:
[KE9] 1126: IF direktwert <= 255 AND direktwert >= Ø TH
           EN GOTO daten.ok
[GA2] 1127: direktwert = Ø
[E2Ø] 1128: '
[6ZC] 1129: IF lauf = 2 THEN fehler = fehler + 1: PRINT
            #2, "==> Fehler: Datenanweisung zu lang"
[1G1] 113Ø: daten.ok:
[E2Ø] 1131: '
[ZX4] 1132: befehllaenge = befehllaenge + 1
[2Z5] 1133: befehltab(befehllaenge) = direktwert
[651] 1134: RETURN
[E2Ø] 1135: '
[FL4] 1136: 'Bildet eine 16-Bit-Adresse
```

```
[EV1] 1137: baut.16bit:
[6DA] 1138: IF zieltyp = speicher THEN lsb = zielwert1
            ELSE lsb = quellwert1
[YM2] 1139: GOSUB teil.16bit
[3Y4] 114Ø: befehllaenge = befehllaenge + 2
[4Ø5] 1141: befehltab(befehllaenge - 1) = lsb
[8M4] 1142: befehltab(befehllaenge) = msb
[E2Ø] 1143: '
[651] 1144: RETURN
[E2Ø] 1145: '
[H43] 1146: 'Pseudobefehl 'EQU'
[Z81] 1147: ps.equ:
[FZ4] 1148: IF (marke$ <> "") THEN GOTO equ.w1
[74H] 1149: IF lauf = 2 THEN fehler = fehler + 1: PRINT
             #2, "==> Fehler: Die Symbolzuweisung fehlt
             bei der EQU-Anweisung"
[651] 115Ø: RETURN
[E2Ø] 1151: '
[891] 1152: equ.w1:
[T85] 1153: IF lauf = 2 THEN GOTO equ.schluss
[XD7] 1154: IF zieltyp <> (nah OR speicher) THEN GOTO e
            qu.w2
[VX2] 1155: fehler = fehler + 1
[ZVE] 1156: PRINT #2, "==> Fehler: EQU betrifft in Zeil
            e"; zeilnum; "eine noch nicht definierte Zu
            weisung"
[651] 1157: RETURN
[E2Ø] 1158: '
[B91] 1159: equ.w2:
[SS4] 116Ø: wert1(aktsymbol) = zielwert1
[D95] 1161: symboltyp(aktsymbol) = zieltyp
[382] 1162: equ.schluss:
[E2Ø] 1163: '
[651] 1164: RETURN
[E2Ø] 1165: '
[L33] 1166: 'Pseudobefehl 'ORG'
[181] 1167: ps.org:
[GB3] 1168: codezeig = zielwert1
[651] 1169: RETURN
[E2Ø] 117Ø: '
[CU2] 1171: 'Pseudobefehl 'DB'
[7YØ] 1172: ps.db:
[754] 1173: IF lauf = 2 THEN GOTO db.w1
[Y88] 1174: IF marke$ <> "" THEN symboltyp(aktsymbol) =
             speicher
[E2Ø] 1175: '
[DVØ] 1176: db.w1:
[447] 1177: zeilzeig1 = opndzeig: zeilzeig2 = opndzeig
[P54] 1178: WHILE zeilzeig1 < endzeig
[4Ø2] 1179:    GOSUB Aufbau
[PZ5] 118Ø:    IF NOT gefunden THEN GOTO db.schluss
[2N4] 1181:    objekt$ = FELD$: GOSUB op.zahl
[5EA] 1182:    IF NOT gefunden OR (zahltyp AND direkt8)
            = falsch THEN GOTO db.w2
[3W1] 1183:    GOSUB db.w4
[BU2] 1184:    GOTO op.bekannt
[E2Ø] 1185:    '
[FVØ] 1186: db.w2:
[UM5] 1187:    IF LEFT$(FELD$, 1) <> "'" THEN GOTO db.w3
[7W1] 1188:    GOSUB db.w5
[BU2] 1189:    GOTO op.bekannt
[E2Ø] 119Ø:    '
[HVØ] 1191: db.w3:
[8GD] 1192:    IF lauf = 2 THEN PRINT #2, "==> Fehler: U
            nbekannter Operand "; FELD$: fehler = fehle
            r + 1
[5Y1] 1193: op.bekannt:
[TRØ] 1194: WEND
[E2Ø] 1195: '
[3Y1] 1196: db.schluss:
[DE5] 1197: codezeig = codezeig + befehllaenge
[651] 1198: RETURN
[E2Ø] 1199: '
[LVØ] 12ØØ: db.w4:
[ZX4] 12Ø1: befehllaenge = befehllaenge + 1
[VP5] 12Ø2: befehltab(befehllaenge) = zahlwert
[651] 12Ø3: RETURN
[E2Ø] 12Ø4: '
[NVØ] 12Ø5: db.w5:
[F45] 12Ø6: FELD$ = MID$(FELD$, 2, LEN(FELD$) - 2)
[E2Ø] 12Ø7: '
[8F3] 12Ø8: FOR i = 1 TO LEN(FELD$)
[ZX4] 12Ø9:    befehllaenge = befehllaenge + 1
[L37] 121Ø:    befehltab(befehllaenge) = ASC(MID$(FELD$,
             i, 1))
[U11] 1211: NEXT i
[E2Ø] 1212: '
[651] 1213: RETURN
[E2Ø] 1214: '
[EV2] 1215: 'Pseudobefehl 'DS'
[92Ø] 1216: ps.ds:
[E12] 1217: dsflag = wahr
[S54] 1218: IF lauf = 2 THEN GOTO ds.w1
[Y88] 1219: IF marke$ <> "" THEN symboltyp(aktsymbol) =
             speicher
[E2Ø] 122Ø: '
[ØXØ] 1221: ds.w1:
[3KD] 1222: IF quelltyp AND direkt8 THEN quellzielwert
            = quellwert1 ELSE quellzielwert = Ø
```

```
[F7A] 1223: IF lauf = 1 THEN GOTO ds.w2 ELSE laenge = l
            aenge + zielwert1 '<- ***
[E2Ø] 1224: '
[XH3] 1225: FOR i = 1 TO zielwert1
[P19] 1226:    LSET byte$ = CHR$(quellzielwert): d.z = d
            .z + 1: PUT #3, d.z
[U11] 1227: NEXT i
[2XØ] 1228: ds.w2:
[E2Ø] 1229: '
[365] 123Ø: codezeig = codezeig + zielwert1
[651] 1231: RETURN
[E2Ø] 1232: '
[393] 1233: 'Pseudobefehl 'PROC'
[SB1] 1234: ps.proc:
[288] 1235: IF stapelzeiger < stapelgroesse THEN GOTO p
            roc.w2
[PC4] 1236: IF lauf = 1 THEN GOTO proc.w1
[E2Ø] 1237: '
[VX2] 1238: fehler = fehler + 1
[DH9] 1239: PRINT #2, "==> Fehler: Prozeduren sind zu t
            ief verschachtelt"
[MB1] 124Ø: proc.w1:
[E2Ø] 1241: '
[651] 1242: RETURN
[E2Ø] 1243: '
[SB1] 1244: proc.w2:
[U65] 1245: stapelzeiger = stapelzeiger + 1
[9E6] 1246: prozedurtyp(stapelzeiger) = zieltyp
[651] 1247: RETURN
[E2Ø] 1248: '
[393] 1249: 'Pseudobefehl 'ENDP'
[SB1] 125Ø: ps.endp:
[XV5] 1251: IF stapelzeiger > Ø THEN GOTO endp.w2
[RB4] 1252: IF lauf = 1 THEN GOTO endp.w1
[E2Ø] 1253: '
[VX2] 1254: fehler = fehler + 1
[1N5] 1255: PRINT #2, "==> Fehler: ENDP ohne PROC"
[E2Ø] 1256: '
[L91] 1257: endp.w1:
[651] 1258: RETURN
[E2Ø] 1259: '
[R91] 126Ø: endp.w2:
[Y65] 1261: stapelzeiger = stapelzeiger - 1
[651] 1262: RETURN
[E2Ø] 1263: '
[EV3] 1264: 'Pseudobefehl 'SEGMENT'
[WZ1] 1265: ps.segment:
[PØ6] 1266: IF lauf = 1 THEN GOTO segment.schluss
[E2Ø] 1267: '
[NL2] 1268: segname$ = marke$
[TWM] 1269: IF segmente = 1 THEN fehler = fehler + 1: P
            RINT "==> Fehler: Unzulässiger Segmentanfan
            g": GOTO segment.schluss ELSE segmente = 1
[A97] 127Ø: pruef$ = MID$(zeile$, INSTR(zeile$, befehl$
            ) + 7)
[E2Ø] 1271: '
[PV2] 1272: segment.w1: '*** ->
[HPA] 1273: IF INSTR(pruef$, ";") THEN pruef$ = LEFT$(p
            ruef$, INSTR(pruef$, ";") - 1)
[5EA] 1274: WHILE INSTR(trenn$, LEFT$(pruef$, 1)): prue
            f$ = MID$(pruef$, 2): WEND
[YGC] 1275: WHILE INSTR(trenn$, RIGHT$(pruef$, 1)): pru
            ef$ = LEFT$(pruef$, LEN(pruef$) - 1): WEND
[SF7] 1276: IF pruef$ = "CODE" THEN code = codezeig: c2
             = wahr
[YEL] 1277: IF c2 AND pruef$ = "STACK" THEN fehler = fe
            hler + 1: PRINT "==> Fehler: Unzulässige Se
            gmentfolge": GOTO segment.schluss
[B88] 1278: IF pruef$ = "STACK" THEN stapel = codezeig:
             s2 = wahr
[PT3] 1279: segment.schluss: '<- ***
[651] 128Ø: RETURN
[E2Ø] 1281: '
[4Z5] 1282: 'Pseudobefehle 'TITEL' und 'SUBTITLE'
[WG1] 1283: ps.title:
[E2Ø] 1284: '
[Ø4B] 1285: IF befehl$ = "TITLE" THEN title$ = MID$(zei
            le$, INSTR(zeile$, befehl$) + 6)
[ØDB] 1286: IF befehl$ = "SUBTTL" THEN subttl$ = MID$(z
            eile$, INSTR(zeile$, befehl$) + 7)
[E2Ø] 1287: '
[651] 1288: RETURN
[E2Ø] 1289: '
[F93] 129Ø: 'Pseudobefehl 'ENDS'
[4C1] 1291: ps.ends:
[E85] 1292: IF lauf = 1 THEN GOTO ens.schluss
[YN5] 1293: IF marke$ = segname$ THEN GOTO ens.w1
[E2Ø] 1294: '
[9PH] 1295: fehler = fehler + 1: PRINT #2, "==> Fehler:
             Das Segmentende hat nicht den gleichen Nam
            en wie der Segmentanfang"
[E2Ø] 1296: '
[R81] 1297: ens.w1:
[S2G] 1298: IF segmente = Ø THEN fehler = fehler + 1: P
            RINT #2, "==> Fehler: Falsches Segmentende"
             ELSE segmente = Ø
[K72] 1299: ens.schluss:
```

Listing 1. »sam3.asc« ist die neue Version 1.1 von SAM III

```
[E2Ø] 13ØØ: '
[651] 13Ø1: RETURN
[E2Ø] 13Ø2: '
[217] 13Ø3: 'Unterprogramm, das den Programmcode und
[Ø15] 13Ø4: ' das Assemblerlisting ausgibt
[Ø52] 13Ø5: code.ausgabe:
[E2Ø] 13Ø6: '
[M5D] 13Ø7: IF dsflag THEN h$ = HEX$(codezeig - zielwer
            t1) ELSE h$ = HEX$(codezeig - befehllaenge)
[KE4] 13Ø8: h$ = STRING$(4 - LEN(h$), "Ø") + h$
[8V5] 13Ø9: IF asslis <> 4 THEN PRINT #2, TAB(1); h$;
[8LØ] 131Ø: i = 1
[2B5] 1311: IF asslis <> 4 THEN PRINT #2, TAB(6);
[E2Ø] 1312: '
[9S1] 1313: WHILE i <= 6
[BA6] 1314:   IF i > befehllaenge THEN GOTO ausgabe.wl
[5S4] 1315:   LSET byte$ = CHR$(befehltab(i))
[LE3] 1316:   d.z = d.z + 1: PUT #3, d.z
[3V2] 1317:   laenge = laenge + 1
[XA3] 1318:   h$ = HEX$(befehltab(i))
[WV3] 1319:   IF LEN(h$) = 1 THEN h$ = "Ø" + h$
[SS4] 132Ø:   IF asslis <> 4 THEN PRINT #2, h$;
[611] 1321:   i = i + 1
[TRØ] 1322: WEND
[E2Ø] 1323: '
[7Y1] 1324: ausgabe.wl:
[BF5] 1325: IF asslis <> 4 THEN PRINT #2, TAB(19);
[TU5] 1326: IF asslis <> 4 THEN PRINT #2, zeilnum;
[E2Ø] 1327: '
[CB9] 1328: IF asslis = 3 THEN PRINT #2, SPACE$(2); zei
            le$: GOTO aus.byte
[VM9] 1329: IF asslis <> 4 THEN PRINT #2, SPACE$(2); LE
            FT$(zeile$, LWIDTH - 26)
[E2Ø] 133Ø: '
[LM1] 1331: aus.byte:
[E2Ø] 1332: '
[RS3] 1333: WHILE i <= befehllaenge
[L18] 1334:   IF i MOD 6 = 1 THEN IF asslis <> 4 THEN P
            RINT #2, TAB(6);
[5S4] 1335:   LSET byte$ = CHR$(befehltab(i))
[LE3] 1336:   d.z = d.z + 1: PUT #3, d.z
[3V2] 1337:   laenge = laenge + 1
[XA3] 1338:   h$ = HEX$(befehltab(i))
[WV3] 1339:   IF LEN(h$) = 1 THEN h$ = "Ø" + h$
[SS4] 134Ø:   IF asslis <> 4 THEN PRINT #2, h$;
[611] 1341:   i = i + 1
[TRØ] 1342: WEND
[E2Ø] 1343: '
[X98] 1344: IF befehllaenge > 6 THEN IF asslis <> 4 THE
            N PRINT #2,
[651] 1345: RETURN
[E2Ø] 1346: '
[BX5] 1347: 'Initialisierung des dritten Laufs
[9R1] 1348: init.drei:
[E2Ø] 1349: '
[Ø51] 135Ø: CLOSE #1
[VØ5] 1351: OPEN "work2.wrk" FOR INPUT AS #1
[Y41] 1352: lauf = 2
[XU1] 1353: codezeig = Ø
[WV1] 1354: zeilnum = Ø
[E2Ø] 1355: '
[651] 1356: RETURN
[E2Ø] 1357: '
[8V5] 1358: 'Beenden der Übersetzungsroutine
[EH2] 1359: assem.schluss:
[E2Ø] 136Ø: '
[8PD] 1361: IF stapelzeiger > Ø THEN fehler = fehler +
            1: PRINT #2, "==> Fehler: Es fehlt ein 'END
            P' "
[UEF] 1362: IF c2 = falsch THEN warnung = warnung + 1:
            PRINT #2, "==> Warnung: Kein Code-Segment s
            pezifiziert"
[6XF] 1363: IF s2 = falsch THEN warnung = warnung + 1:
            PRINT #2, "==> Warnung: Kein Stapel-Segment
             spezifiziert"
[T9E] 1364: IF segmente = 1 THEN fehler = fehler + 1: P
            RINT #2, "==> Fehler: Kein Segmentende ange
            geben"
[E2Ø] 1365: '
[691] 1366: PRINT #2,
[691] 1367: PRINT #2,
[C86] 1368: PRINT #2, fehler; "Fehler festgestellt."
[237] 1369: PRINT #2, warnung; "Warnung(en) ausgegeben.
            "
[E2Ø] 137Ø: '
[862] 1371: GOSUB sym.aus
[E2Ø] 1372: '
[B1A] 1373: PRINT #2, : PRINT #2, laenge; "Bytes für di
            e EXE-Datei erzeugt."
[8N6] 1374: IF l$ = "lpt1:" THEN PRINT #2, CHR$(27); CH
            R$(64)
[3WØ] 1375: CLOSE
[E2Ø] 1376: '
[S26] 1377: IF fehler = Ø THEN GOSUB exe.struktur
[KF2] 1378: SHELL "del *.wrk"
[RNØ] 1379: BEEP
[GXØ] 138Ø: PRINT
[GXØ] 1381: PRINT
[ND6] 1382: PRINT "SAM III hat seine Arbeit beendet."
[GXØ] 1383: PRINT
[E2Ø] 1384: '
[4MØ] 1385: END
[E2Ø] 1386: '
[PD4] 1387: 'Ausgabe der Symboltabelle
[UG1] 1388: sym.aus:
[MX2] 1389: PRINT #2, : PRINT #2,
[8Z3] 139Ø: PRINT #2, "Symboltabelle:"
[HK3] 1391: PRINT #2, "=============="
[E2Ø] 1392: '
[L41] 1393: i = 29 + 1
[MC3] 1394: perzeil = LWIDTH \ 19
[BRE] 1395: IF i > aktsymbol THEN PRINT #2, : PRINT #2,
             "Es wurden keine Symbole definiert.": GOTO
             sym.wl
[E2Ø] 1396: '
[L93] 1397: WHILE i <= aktsymbol
[VP2] 1398:   h$ = HEX$(wert1(i))
[KE4] 1399:   h$ = STRING$(4 - LEN(h$), "Ø") + h$
[BT4] 14ØØ:   PRINT #2, symbol$(i); " "; h$; "     ";
[611] 14Ø1:   i = i + 1
[656] 14Ø2:   IF (i - 29) MOD perzeil = 1 THEN PRINT #2
              ,
[TRØ] 14Ø3: WEND
[E2Ø] 14Ø4: '
[W91] 14Ø5: sym.wl:
[691] 14Ø6: PRINT #2,
[651] 14Ø7: RETURN
[E2Ø] 14Ø8: '
[KE5] 14Ø9: 'Angabe der aktuellen Parameter
[FT1] 141Ø: aus.param:
[E2Ø] 1411: '
[XD3] 1412: X = POS(Ø): Y = CSRLIN
[HA3] 1413: LOCATE 25, 1: COLOR Ø, 7
[2WE] 1414: PRINT "Durchlauf"; lauf + 2; "ist aktiv.
                                Anzahl der bearbeiteten Ze
            ilen: "; zeilnum;
[GG3] 1415: COLOR 7, Ø: LOCATE Y, X
[E2Ø] 1416: '
[651] 1417: RETURN
[E2Ø] 1418: '
[375] 1419: 'Unterprogramm für EXE-Dateien
[2D2] 142Ø: exe.struktur:
[E29] 1421: OPEN "work.wrk" FOR RANDOM AS #3 LEN = 1: F
            IELD #3, 1 AS byte$
[XY8] 1422: OPEN zieldat$ FOR RANDOM AS #2 LEN = 1: FIE
            LD #2, 1 AS ZBYTE$
[M11] 1423: d.z = Ø
[E2Ø] 1424: '
[6X6] 1425: LSET ZBYTE$ = CHR$(77): d.z = d.z + 1: PUT
            #2, d.z
[UW6] 1426: LSET ZBYTE$ = CHR$(9Ø): d.z = d.z + 1: PUT
            #2, d.z
[D3A] 1427: LSET ZBYTE$ = CHR$((laenge MOD 512) MOD 256
            ): d.z = d.z + 1: PUT #2, d.z
[PCA] 1428: LSET ZBYTE$ = CHR$(INT((laenge MOD 512) / 2
            56)): d.z = d.z + 1: PUT #2, d.z
[AYC] 1429: LSET ZBYTE$ = CHR$(INT(laenge / 512) + 1 +
            SGN(laenge MOD 512) * 1): d.z = d.z + 1: PU
            T #2, d.z
[HH6] 143Ø: LSET ZBYTE$ = CHR$(Ø): d.z = d.z + 1: PUT #
            2, d.z
[L1A] 1431: LSET ZBYTE$ = CHR$(Ø): d.z = d.z + 1: PUT #
            2, d.z: d.z = d.z + 1: PUT #2, d.z
[SW6] 1432: LSET ZBYTE$ = CHR$(32): d.z = d.z + 1: PUT
            #2, d.z
[NHD] 1433: LSET ZBYTE$ = CHR$(Ø): d.z = d.z + 1: PUT #
            2, d.z: d.z = d.z + 1: PUT #2, d.z: d.z = d
            .z + 1: PUT #2, d.z
[WZA] 1434: LSET ZBYTE$ = CHR$(255): d.z = d.z + 1: PUT
             #2, d.z: d.z = d.z + 1: PUT #2, d.z
[L1A] 1435: LSET ZBYTE$ = CHR$(Ø): d.z = d.z + 1: PUT #
            2, d.z: d.z = d.z + 1: PUT #2, d.z
[E2Ø] 1436: '
[3E3] 1437: 'Nun das SP-Register
[KH9] 1438: LSET ZBYTE$ = CHR$(stapel MOD 256): d.z = d
            .z + 1: PUT #2, d.z '<- ***
[HN9] 1439: LSET ZBYTE$ = CHR$(INT(stapel / 255)): d.z
            = d.z + 1: PUT #2, d.z '<- ***
[667] 144Ø: LSET ZBYTE$ = CHR$(237): d.z = d.z + 1: PUT
             #2, d.z
[YW6] 1441: LSET ZBYTE$ = CHR$(54): d.z = d.z + 1: PUT
            #2, d.z
[E2Ø] 1442: '
[5D3] 1443: 'Nun das IP-Register
[6E9] 1444: LSET ZBYTE$ = CHR$(code MOD 256): d.z = d.z
             + 1: PUT #2, d.z '<- ***
[VR9] 1445: LSET ZBYTE$ = CHR$(INT(code / 255)): d.z =
            d.z + 1: PUT #2, d.z '<- ***
[L1A] 1446: LSET ZBYTE$ = CHR$(Ø): d.z = d.z + 1: PUT #
            2, d.z: d.z = d.z + 1: PUT #2, d.z
[MW6] 1447: LSET ZBYTE$ = CHR$(3Ø): d.z = d.z + 1: PUT
            #2, d.z
[NHD] 1448: LSET ZBYTE$ = CHR$(Ø): d.z = d.z + 1: PUT #
            2, d.z: d.z = d.z + 1: PUT #2, d.z: d.z = d
            .z + 1: PUT #2, d.z
[KH6] 1449: LSET ZBYTE$ = CHR$(1): d.z = d.z + 1: PUT #
            2, d.z
[F13] 145Ø: LSET ZBYTE$ = CHR$(Ø)
```

```
[E2Ø] 1451: '
[R92] 1452: FOR i = 1 TO 483
[HE3] 1453:   d.z = d.z + 1: PUT #2, d.z
[U11] 1454: NEXT i
[E2Ø] 1455: '
[U13] 1456: FOR i = 1 TO laenge
[E81] 1457:   GET #3, i
[ZA1] 1458:   a$ = byte$
[HE2] 1459:   LSET ZBYTE$ = a$
[HE3] 1460:   d.z = d.z + 1: PUT #2, d.z
[U11] 1461: NEXT i
[E2Ø] 1462: '
[F51] 1463: CLOSE 3
[B51] 1464: CLOSE 2
[E2Ø] 1465: '
[651] 1466: RETURN
[E2Ø] 1467: '
[346] 1468: 'Initialisierung der flexiblen Daten
[AR1] 1469: flex.init:
[YE3] 147Ø: title$ = "": subtt1$ = ""
[M11] 1471: d.z = Ø
[E2Ø] 1472: '
[KC8] 1473: operand = 1: konst2 = 4: regsum = 64: erwei
            bedarf = 128
[W3C] 1474: disp8bedarf = 256: disp16bedarf = 512: modb
            ytebedarf = 1Ø24: direkt8bedarf = 2Ø48
[ERB] 1475: direkt16bedarf = 4Ø96: richtung = 8192: spe
            icherbedarf = 16384: konst1 = &H8ØØØ
[TK9] 1476: akku8 = 1: akku16 = 2: reg8 = 4: reg16 = 8:
             speicherreg = 16: cs = 32
[3KA] 1477: segreg = 64: speicher = 128: direkt8 = 256:
             direkt16 = 512: nichts = 1Ø24
[Y67] 1478: zeichen = 2Ø48: nah = 4Ø96: weit = 8192: cl
            = 16384
[E2Ø] 1479: '
[5M2] 148Ø: stapelzeiger = Ø
[1R8] 1481: code = Ø: stapel = Ø: segmente = Ø: c2 = fa
            lsch: s2 = falsch
[DK3] 1482: fehler = Ø: warnung = Ø
[7D1] 1483: laenge = Ø
[E2Ø] 1484: '
[Z96] 1485: 'Öffnen der Ein- und Ausgabedateien
[WT4] 1486: OPEN quelldat$ FOR INPUT AS #1
[E2Ø] 1487: '
[TLG] 1488: IF asslis = 3 THEN l$ = LEFT$(zieldat$, LEN
            (zieldat$) - 3) + "lst": GOTO flex.oeffnen:
             'Ausgabe in eine Datei
[FGA] 1489: IF asslis = 2 THEN l$ = "lpt1:": GOTO flex.
            oeffnen: 'Druckerausgabe
[EM1] 149Ø: l$ = "scrn:"
[E2Ø] 1491: '
[B62] 1492: flex.oeffnen:
[E2Ø] 1493: '
[DS3] 1494: OPEN l$ FOR OUTPUT AS #2
[2W1] 1495: O$ = zieldat$
[ØF3] 1496: PRINT : PRINT : PRINT
[E2Ø] 1497: '
[ZZ5] 1498: OPEN "work.wrk" FOR RANDOM AS #3 LEN = 1
[DT2] 1499: FIELD #3, 1 AS byte$
[BS1] 15ØØ: LWIDTH = 79
[BS8] 15Ø1: C$ = MID$(DATE$, 4, 2) + "." + LEFT$(DATE$,
             2) + "." + RIGHT$(DATE$, 4)
[DDB] 15Ø2: IF asslis <> 4 THEN PRINT #2, quelldat$; TA
            B(LWIDTH - LEN(C$)); C$: PRINT #2,
[Y74] 15Ø3: IF asslis <> 4 THEN PRINT #2,
[RUD] 15Ø4: IF asslis <> 4 THEN PRINT #2, "Adr."; TAB(6
            ); "Hex-Code"; TAB(18); "Zeile"; TAB(25); "
            Quelltext"
[Y74] 15Ø5: IF asslis <> 4 THEN PRINT #2,
[E2Ø] 15Ø6: '
[651] 15Ø7: RETURN
[E2Ø] 15Ø8: '
[ZE7] 15Ø9: '*** -> Fehlerabfangroutine für das Laden d
            er
[X33] 151Ø: ' Datei 'SAM3V11.DAT'
[912] 1511: samdat.fehl:
[E2Ø] 1512: '
[SXF] 1513: PRINT : PRINT : PRINT "   SAM III bricht ab
            , weil die Datei 'SAM3V11.DAT' nicht gefund
            en werden kann."
[GUØ] 1514: '<- ***
[4MØ] 1515: END
[E2Ø] 1516: '
[XØ6] 1517: 'Fehlerabfangroutine für 'INCLUDE'
[T62] 1518: fehl.include:
[FS4] 1519: IF NOT DOS THEN GOTO fehl.inwl
[MHE] 152Ø: PRINT : PRINT "SAM III wurde durch einen Fe
            hler beim INCLUDE-Befehl unterbrochen.": PR
            INT
[4MØ] 1521: END
[E2Ø] 1522: '
[XM1] 1523: fehl.inwl:
[BPU] 1524: LOCATE 1Ø, 6: PRINT "Sie haben den INCLUDE-
            Befehl falsch benutzt und müssen Ihre Einga
            be": LOCATE 12, 14: PRINT "berichtigen. Drü
            cken Sie bitte eine beliebige Taste !"
[5H3] 1525: WHILE INKEY$ = "": WEND
[3WØ] 1526: CLOSE
[WS1] 1527: GOTO menue
[E2Ø] 1528: '
[YR5] 1529: 'Fehlerabfangroutine für 'MACRO'
[E42] 153Ø: macro.fehler:
[ZA4] 1531: IF NOT DOS THEN GOTO mac.wl
[F6E] 1532: PRINT : PRINT "SAM III wurde durch einen Fe
            hler beim MACRO-Befehl unterbrochen.": PRIN
            T
[4MØ] 1533: END
[E2Ø] 1534: '
[Ø61] 1535: mac.wl:
[ØUS] 1536: PRINT : PRINT "Sie haben den MACRO-Befehl f
            alsch benutzt und müssen Ihre Eingabe beric
            htigen.": PRINT : PRINT "Drücken Sie bitte
            eine beliebige Taste !"
[5H3] 1537: WHILE INKEY$ = "": WEND
[3WØ] 1538: CLOSE
[WS1] 1539: GOTO menue
[E2Ø] 154Ø: '
[R25] 1541: 'Fehlerabfangroutine für 'DO'
[WM1] 1542: do.fehler:
[344] 1543: IF NOT DOS THEN GOTO do.wl
[TGD] 1544: PRINT : PRINT "SAM III wurde durch einen Fe
            hler beim DO-Befehl unterbrochen.": PRINT
[4MØ] 1545: END
[E2Ø] 1546: '
[MWØ] 1547: do.wl:
[HØR] 1548: PRINT : PRINT "Der DO-Befehl wurde falsch b
            enutzt. Sie müssen den Befehl berichtigen."
            : PRINT : PRINT "Drücken Sie bitte eine bel
            iebige Taste !"
[5H3] 1549: WHILE INKEY$ = "": WEND
[3WØ] 155Ø: CLOSE
[WS1] 1551: GOTO menue
[E2Ø] 1552: '
[TU6] 1553: 'Erster Lauf (bereitet die Verwendung von
[4W6] 1554: ' Makros und deren Parameterübergabe vor)
[6R1] 1555: lauf.eins:
[ZM4] 1556: IF makros THEN GOSUB inc.exm
[HK5] 1557: OPEN "makros.wrk" FOR OUTPUT AS #6
[4E5] 1558: OPEN "work5.wrk" FOR OUTPUT AS #4
[E2Ø] 1559: '
[4N3] 156Ø: lauf = -1: aktsymbol = 29
[XU1] 1561: codezeig = Ø
[WV1] 1562: zeilnum = Ø
[E2Ø] 1563: '
[3E2] 1564: WHILE NOT EOF(1)
[SL6] 1565:   GOSUB lies.zeile: 'Einlesen einer Zeile
[YR5] 1566:   GOSUB analyse: 'Zeile übersetzen
[E2Ø] 1567:   '
[BR6] 1568:   'Falls Makro-Anweisungen vorhanden sind,
[G27] 1569:   ' so werden sie registriert und für die
[6S5] 157Ø:   ' spätere Verwendung vorbereitet
[KY5] 1571:   IF befehl$ <> "MACRO" THEN GOTO defp.ok
[TX2] 1572:   MACRONAME$ = marke$
[42H] 1573:   IF marke$ = "" THEN PRINT : PRINT "Das Ma
            kro ist nicht durch einen Namen gekennzeich
            net.": GOTO macro.fehler
[WZ7] 1574:   Parameter$ = MID$(zeile$, INSTR(zeile$, b
            efehl$) + 6)
[E2Ø] 1575:   '
[ME4] 1576:   WHILE INSTR(Parameter$, ";")
[ML8] 1577:     Parameter$ = LEFT$(Parameter$, INSTR(Pa
            rameter$, ";") - 1)
[TRØ] 1578:   WEND
[E2Ø] 1579:   '
[1K4] 158Ø:   WHILE LEFT$(Parameter$, 1) = " "
[LZ4] 1581:     Parameter$ = MID$(Parameter$, 2)
[TRØ] 1582:   WEND
[E2Ø] 1583:   '
[BU4] 1584:   WHILE RIGHT$(Parameter$, 1) = " "
[6M7] 1585:     Parameter$ = LEFT$(Parameter$, LEN(Para
            meter$) - 1)
[TRØ] 1586:   WEND
[E2Ø] 1587:   '
[R33] 1588:   werte = Ø: werte$ = ""
[262] 1589: param.falsch:
[Y2M] 159Ø:   IF werte = 9 THEN PRINT : PRINT "Zu viele
             oder falsche Parameter bei der Definition
            des Makros: "; marke$: GOTO macro.fehler
[9E6] 1591:   IF Parameter$ = werte$ THEN GOTO mdef.anf
[P49] 1592:   IF werte = Ø THEN werte = 1: werte$ = "%1
            ": GOTO param.falsch
[8K2] 1593:   werte = werte + 1
[FP6] 1594:   werte$ = werte$ + ",%" + RIGHT$(STR$(wert
            e), 1)
[CW2] 1595:   GOTO param.falsch
[E2Ø] 1596:   '
[HE1] 1597: mdef.anf:
[FL5] 1598:   PRINT #6, "Makrodefinition: Anfang"
[ZE2] 1599:   PRINT #6, werte
[WD2] 16ØØ:   PRINT #6, marke$
[E2Ø] 16Ø1:   '
[LR1] 16Ø2: enm.fehlt:
[BEF] 16Ø3:   IF EOF(1) THEN PRINT : PRINT "Der Befehl
            'ENDM' fehlt bei dem Makro: "; MACRONAME$:
            GOTO macro.fehler
```

Listing 1. »sam3.asc« ist die neue Version 1.1 von SAM III

```
[SL6] 16Ø4:   GOSUB lies.zeile: 'Einlesen einer Zeile
[YR5] 16Ø5:   GOSUB analyse: 'Zeile übersetzen
[E2Ø] 16Ø6:   '
[ØEB] 16Ø7:   IF befehl$ = "ENDM" THEN PRINT #6, "Makro
          definition: Ende": GOTO def.param
[5F2] 16Ø8:   PRINT #6, zeile$
[CE2] 16Ø9:   GOTO enm.fehlt
[E2Ø] 161Ø:   '
[HP1] 1611: def.param:
[ME1] 1612:   zeile$ = ""
[E2Ø] 1613:   '
[882] 1614:   'Weitermachen
[EB1] 1615: defp.ok:
[5N2] 1616:   GOSUB aus.param
[285] 1617:   IF zeile$ <> "" THEN PRINT #4, zeile$
[TRØ] 1618: WEND
[E2Ø] 1619: '
[A14] 162Ø: CLOSE #4: CLOSE #6: CLOSE #1
[E2Ø] 1621: '
[651] 1622: RETURN
[E2Ø] 1623: '
[D36] 1624: 'Zweiter Include-Lauf zum Einbinden
[MS2] 1625: ' externer Makros
[PD1] 1626: inc.exm:
[LU7] 1627: codezeig = Ø: zeilnum = Ø: lauf = -2: aktsy
          mbol = 29
[8E5] 1628: OPEN "work6.wrk" FOR OUTPUT AS #4
[E2Ø] 1629: '
[3E2] 163Ø: WHILE NOT EOF(1)
[8R2] 1631:   GOSUB lies.zeile
[T72] 1632:   GOSUB analyse
[BW6] 1633:   IF befehl$ <> "INCLUDE" THEN GOTO inc.sch
          luss
[277] 1634:   datei$ = MID$(zeile$, INSTR(zeile$, befeh
          l$) + 8)
[494] 1635:   WHILE INSTR(datei$, ";") <> Ø
[4B6] 1636:     datei$ = LEFT$(datei$, INSTR(datei$, ";
          ") - 1)
[TRØ] 1637:   WEND
[5T3] 1638:   WHILE LEFT$(datei$, 1) = " "
[3D3] 1639:     datei$ = MID$(datei$, 2)
[TRØ] 164Ø:   WEND
[X14] 1641:   WHILE RIGHT$(datei$, 1) = " "
[YC5] 1642:     datei$ = LEFT$(datei$, LEN(datei$) - 1)
[TRØ] 1643:   WEND
[R94] 1644:   OPEN datei$ FOR INPUT AS #5
[LE2] 1645:   WHILE NOT EOF(5)
[LE2] 1646:     LINE INPUT #5, a$
[GN1] 1647:     PRINT #4, a$
[TRØ] 1648:   WEND
[451] 1649:   CLOSE #5
[ME1] 165Ø:   zeile$ = ""
[U52] 1651: inc.schluss:
[5N2] 1652:   GOSUB aus.param
[285] 1653:   IF zeile$ <> "" THEN PRINT #4, zeile$
[TRØ] 1654: WEND
[E2Ø] 1655: '
[351] 1656: CLOSE #4
[Ø51] 1657: CLOSE #1
[E2Ø] 1658: '
[B15] 1659: OPEN "work6.wrk" FOR INPUT AS #1
[651] 166Ø: RETURN
```

Listing 1. »sam3.asc« ist die neue Version 1.1 von SAM III

```
[L36]    1: prog segment code  ;Programmsegment
[114]    2:       org     1ØØh  ;für COM-Datei
[L14]    3:                     ;entfällt bei EXE-Datei
[3G3]    4:       include  macrobib.sam
[A93]    5:                     ;MACROBIB einbinden
[HP3]    6:                     ;INCLUDE darf nicht in
[7L3]    7:                     ;erster Spalte stehen
[RU1]    8:       do   clrscr
[6Ø3]    9:       do   setcursor,4Ø,5
[4W2]   1Ø:       do   zeichenaus,65
[913]   11:       do   zeichenaus,'B'
[5P3]   12:       do   Zeichenaus,Ø1ØØØØ11B
[SZ2]   13:       do   setcursor,Ø,2Ø
[ZN4]   14:       do   textaus,offset(text),cs
[UZ2]   15:       do   setcursor,5,1Ø
[BS1]   16:       mov  al,'A'
[PC1]   17:       push ax
[ZY2]   18:       do   zeichenaus,al
[UØ3]   19:       do   zeichenaus,'='
[R31]   2Ø:       pop ax
[S72]   21:       do   hexaus,al
[2Ø3]   22:       do   setcursor,5,12
[HA3]   23:       do   hexwortaus,1234h
[AØ3]   24:       do   setcursor,5,14
[HS1]   25:       mov  ax,es
[L23]   26:       do   hexwortaus,ax
[DS1]   27:       mov  ax,cs
[263]   28:       do   h exwortaus,ax
[FS1]   29:       mov  ax,ds
[L23]   3Ø:       do   hexwortaus,ax
[DT1]   31:       mov  ax,ss
[L23]   32:       do   hexwortaus,ax
[SY1]   33:       do   schluss
[ØØØ]   34:
```

```
[C7B]   35: text db 'Macrobib enthält: Clrscr, Zeichena
          us, Setcursor, Textaus,'
[9D8]   36:       db Øah,Ødh,'Hexaus, Hexwortaus, Pushre
          gs, Popregs.'
[CT1]   37:       db Øah,Ødh,'$'
[AK1]   38: prog ends
[ØØØ]   39:
[4MØ]   4Ø:       end
```

Listing 2. »mbibdemo.sam« ist ein Beispielprogramm für die Verwendung von Makros

```
[T38]    1: ; MACROBIB.SAM einbinden mit: INCLUDE MACRO
          BIB.SAM
[P98]    2: ; INCLUDE darf nicht in der ersten Spalte b
          eginnen!
[1RA]    3: ; Makroaufruf mit:  DO MACRONAME[,Parameter
          1][,Parameter2][,...]
[ØØØ]    4:
[6F6]    5: clrscr   macro       ;Bildschirm löschen
[DM4]    6:          mov ah,Øfh ;Videomodus lesen
[Y71]    7:          int 1Øh
[UN4]    8:          mov ah,Ø    ;Videomodus setzen
[Y71]    9:          int 1Øh
[HPØ]   1Ø:          endm
[ØØØ]   11:
[K37]   12: textaus  macro %1,%2 ;Zeichenkette ausgeben
[WT4]   13:          mov ax,%2   ;auf Standardgerät
[5F5]   14:          mov ds,ax   ;Segment Zeichenkette
[PB5]   15:          mov dx,%1   ;Adresse Zeichenkette
[HB3]   16:          mov ah,9    ;Ende mit '$'
[381]   17:          int 21h
[HPØ]   18:          endm
[ØØØ]   19:
[S16]   2Ø: zeichenaus macro %1 ;Zeichen ausgeben
[GR3]   21:          mov dl,%1   ; Zeichencode
[V91]   22:          mov ah,2
[381]   23:          int 21h
[HPØ]   24:          endm
[ØØØ]   25:
[116]   26: setcursor macro %1,%2 ;Cursor setzen
[DM4]   27:          mov ah,Øfh ;Videomodus lesen
[Y71]   28:          int 1Øh
[NL1]   29:          mov ah,Ø2h
[US2]   3Ø:          mov dl,%1   ;Spalte
[GH2]   31:          mov dh,%2   ;Zeile
[FY3]   32:          int 1Øh     ;Cursor setzen
[HPØ]   33:          endm
[ØØØ]   34:
[9F2]   35: schluss  macro
[P55]   36:          mov ax,Ø4cØØh ;Programm beenden
[381]   37:          int 21h
[HPØ]   38:          endm
[ØØØ]   39:
[247]   4Ø: hexaus   macro %1    ;Byte als Hexzahl ausge
          ben
[WS4]   41:          mov dl,%1   ;auf Standardgerät
[ZC1]   42:          push dx
[RW4]   43:          shr dl      ;Sprungbefehle als DB
[GX4]   44:          shr dl      ;codiert, weil keine
[HU5]   45:          shr dl      ;lokalen Marken verwalt
          et
[AY3]   46:          shr dl      ;werden können
[CZ4]   47:          db e8h,Ø6h,ØØh ;call Hexausnibble
[X31]   48:          pop dx
[4Z4]   49:          db e8h,Ø2h,ØØh ;call Hexausnibble
[C33]   5Ø:          db ebh,13h ;jmps Ende
[TR4]   51:          proc near  ;hier Hexausnibble
[CK1]   52:          and dl,Øfh
[Ø81]   53:          or dl,3Øh
[PL1]   54:          cmp dl,3ah
[863]   55:          db 7ch,Ø3h ;jl Ausgabe
[GG1]   56:          add dl,Ø7h
[FN3]   57:          mov ah,2    ;hier Ausgabe
[381]   58:          int 21h
[3PØ]   59:          ret
[MPØ]   6Ø:          endp
[KU2]   61:          endm        ;hier Ende
[ØØØ]   62:
[CM7]   63: hexwortaus macro %1 ;Wort als Hexzahl ausge
          ben
[ØU4]   64:          mov dx,%1   ;auf Standardgerät
[ZC1]   65:          push dx
[FF1]   66:          mov dl,dh
[E25]   67:          db e8h,Ø6h,ØØh ;call Hexwortausm1
[X31]   68:          pop dx
[625]   69:          db e8h,Ø2h,ØØh ;call Hexwortausm1
[YE4]   7Ø:          db ebh,24h ;jmps Hexwortende
[VU4]   71:          proc near  ;hier Hexwortausm1
[ZC1]   72:          push dx
[K21]   73:          shr dl
[K21]   74:          shr dl
[K21]   75:          shr dl
[K21]   76:          shr dl
[E25]   77:          db e8h,Ø5h,ØØh ;call hexwortausm2
[X31]   78:          pop dx
[625]   79:          db e8h,Ø1h,ØØh ;call hexwortausm2
[3PØ]   8Ø:          ret
```

```
[MPØ]    81:            endp
[XU4]    82:            proc near  ;hier Hexwortausm2
[CK1]    83:            and dl,Øfh
[Ø81]    84:            or dl,3Øh
[PL1]    85:            cmp dl,3ah
[863]    86:            db 7ch,Ø3h ;jl Ausgabe
[GG1]    87:            add dl,Ø7h
[FN3]    88:            mov ah,2   ;hier Ausgabe
[381]    89:            int 21h
[3PØ]    9Ø:            ret
[MPØ]    91:            endp
[164]    92:            endm       ;hier Hexwortende
[ØØØ]    93:
[GF6]    94: pushregs macro      ;Register auf Stapel-
[N54]    95:            push si    ;speicher legen
[3B1]    96:            push di
[TB1]    97:            push bp
[ZC1]    98:            push dx
```

```
[WC1]    99:            push cx
[TC1]   1ØØ:            push bx
[PC1]   1Ø1:            push ax
[CD1]   1Ø2:            push sp
[HPØ]   1Ø3:            endm
[ØØØ]   1Ø4:
[3H6]   1Ø5: popregs    macro      ;Register von Stapel-
[M14]   1Ø6:            pop sp     ;speicher lesen
[R31]   1Ø7:            pop ax
[T31]   1Ø8:            pop bx
[V31]   1Ø9:            pop cx
[X31]   11Ø:            pop dx
[131]   111:            pop bp
[G21]   112:            pop di
[E31]   113:            pop si
[HPØ]   114:            endm
```

Listing 3. Die Makro-Bibliothek »macrobib.sam« für Listing 3

```
[3R1]     1: =,1,896,1Ø24,Ø
[A92]     2: AAA,55,1Ø24,1Ø24,1
[BF2]     3: AAD,213,1Ø24,1Ø24,1
[BG2]     4: AAM,212,1Ø24,1Ø24,1
[EB2]     5: AAS,63,1Ø24,1Ø24,1
[M52]     6: ADC,2Ø,3,768,41Ø1
[NH2]     7: ADC,128,12,768,4245
[V92]     8: ADC,16,144,12,1Ø29
[AA2]     9: ADC,18,12,144,9221
[ØP2]    1Ø: ADCB,128,144,256,4241
[BR2]    11: ADCW,129,144,768,4241
[P32]    12: ADD,4,3,768,41Ø1
[3K2]    13: ADD,128,12,768,4229
[R42]    14: ADD,Ø,144,12,1Ø29
[P52]    15: ADD,2,12,144,9221
[AP2]    16: ADDB,128,144,256,4225
[NR2]    17: ADDW,129,144,768,4225
[A72]    18: AND,36,3,768,41Ø1
[DK2]    19: AND,128,12,768,4261
[NA2]    2Ø: AND,32,144,12,1Ø29
[4B2]    21: AND,34,12,144,9221
[FR2]    22: ANDB,128,144,256,4257
[US2]    23: ANDW,129,144,768,4257
[DT2]    24: CALL,154,768,768,4Ø97
[2V2]    25: CALL,232,4Ø96,1Ø24,513
[WV2]    26: CALLF,255,144,1Ø24,153
[GW2]    27: CALLN,255,144,1Ø24,145
[4K2]    28: CBW,152,1Ø24,1Ø24,1
[9H2]    29: CLC,248,1Ø24,1Ø24,1
[SG2]    3Ø: CLD,252,1Ø24,1Ø24,1
[4H2]    31: CLI,25Ø,1Ø24,1Ø24,1
[ØH2]    32: CMC,245,1Ø24,1Ø24,1
[F82]    33: CMP,6Ø,3,768,41Ø1
[MM2]    34: CMP,128,12,768,4285
[PC2]    35: CMP,56,144,12,1Ø29
[5D2]    36: CMP,58,12,144,9221
[RS2]    37: CMPB,128,144,256,4281
[VS2]    38: CMPSB,166,1Ø24,1Ø24,1
[7U2]    39: CMPSW,167,1Ø24,1Ø24,1
[2U2]    4Ø: CMPW,129,144,768,4281
[VH2]    41: CWD,153,1Ø24,1Ø24,1
[S92]    42: DAA,39,1Ø24,1Ø24,1
[WB2]    43: DAS,47,1Ø24,1Ø24,1
[BC2]    44: DB,3,23Ø4,&HFFFF,Ø
[UA2]    45: DEC,254,4,1Ø24,137
[H22]    46: DEC,72,8,1Ø24,65
[SN2]    47: DECB,254,144,1Ø24,137
[EP2]    48: DECW,255,144,1Ø24,137
[LE2]    49: DIV,246,3,144,8373
[1Ø2]    5Ø: DS,4,768,128Ø,Ø
[M33]    51: END,9,&HFFFF,&HFFFF,Ø
[S73]    52: ENDP,6,&HFFFF,&HFFFF,Ø
[SF3]    53: ENDS,11,&HFFFF,&HFFFF,Ø
[Ø62]    54: EQU,1,896,1Ø24,Ø
[S83]    55: EVEN,8,&HFFFF,&HFFFF,Ø
[7L2]    56: HLT,244,1Ø24,1Ø24,1
[WK2]    57: IDIV,246,3,144,8381
[7N2]    58: IMUL,246,3,144,8365
[8H1]    59: IN,236,1,8,1
[CH1]    6Ø: IN,237,2,8,1
[E42]    61: IN,228,3,256,2Ø53
[3C2]    62: INC,254,4,1Ø24,129
[U32]    63: INC,64,8,1Ø24,65
[ØR2]    64: INCB,254,144,1Ø24,129
[PR2]    65: INCW,255,144,1Ø24,129
[3L2]    66: INT,2Ø4,1Ø24,1Ø24,1
[NT2]    67: INT,2Ø5,256,1Ø24,2Ø49
[XM2]    68: INTO,2Ø6,1Ø24,1Ø24,1
[LL2]    69: IRET,2Ø7,1Ø24,1Ø24,1
[ZG2]    7Ø: JA,119,4Ø96,1Ø24,257
[KP2]    71: JAE,115,4Ø96,1Ø24,257
[LG2]    72: JB,114,4Ø96,1Ø24,257
[1R2]    73: JBE,118,4Ø96,1Ø24,257
[PG2]    74: JC,114,4Ø96,1Ø24,257
[6Y2]    75: JCXZ,227,4Ø96,1Ø24,257
[2H2]    76: JE,116,4Ø96,1Ø24,257
[DH2]    77: JG,127,4Ø96,1Ø24,257
[7R2]    78: JGE,125,4Ø96,1Ø24,257
[LH2]    79: JL,124,4Ø96,1Ø24,257
[UR2]    8Ø: JLE,126,4Ø96,1Ø24,257
[BT2]    81: JMP,234,768,768,4Ø97
[2T2]    82: JMP,233,4Ø96,1Ø24,513
[WS2]    83: JMPF,255,144,1Ø24,169
[MS2]    84: JMPN,255,144,1Ø24,161
[ZX2]    85: JMPS,235,4Ø96,1Ø24,257
[NR2]    86: JNA,118,4Ø96,1Ø24,257
[CV2]    87: JNAE,114,4Ø96,1Ø24,257
[DR2]    88: JNB,115,4Ø96,1Ø24,257
```

```
[NV2]    89: JNBE,119,4Ø96,1Ø24,257
[HR2]    9Ø: JNC,115,4Ø96,1Ø24,257
[1S2]    91: JNE,117,4Ø96,1Ø24,257
[8S2]    92: JNG,126,4Ø96,1Ø24,257
[8W2]    93: JNGE,124,4Ø96,1Ø24,257
[SS2]    94: JNL,125,4Ø96,1Ø24,257
[ZW2]    95: JNLE,127,4Ø96,1Ø24,257
[TS2]    96: JNO,113,4Ø96,1Ø24,257
[ØT2]    97: JNP,123,4Ø96,1Ø24,257
[4T2]    98: JNS,121,4Ø96,1Ø24,257
[NU2]    99: JNZ,117,4Ø96,1Ø24,257
[MH2]   1ØØ: JO,112,4Ø96,1Ø24,257
[TH2]   1Ø1: JP,122,4Ø96,1Ø24,257
[PR2]   1Ø2: JPE,122,4Ø96,1Ø24,257
[2T2]   1Ø3: JPO,123,4Ø96,1Ø24,257
[WH2]   1Ø4: JS,12Ø,4Ø96,1Ø24,257
[1L2]   1Ø5: JZ,116,4Ø96,1Ø24,257
[RK2]   1Ø6: LAHF,159,1Ø24,1Ø24,1
[6F2]   1Ø7: LDS,197,8,144,9217
[UB2]   1Ø8: LEA,141,8,144,9217
[5F2]   1Ø9: LES,196,8,144,9217
[8L2]   11Ø: LOCK,24Ø,1Ø24,1Ø24,1
[WR2]   111: LODSB,172,1Ø24,1Ø24,1
[8T2]   112: LODSW,173,1Ø24,1Ø24,1
[ZX2]   113: LOOP,226,4Ø96,1Ø24,257
[D63]   114: LOOPE,225,4Ø96,1Ø24,257
[S93]   115: LOOPNE,224,4Ø96,1Ø24,257
[RB3]   116: LOOPNZ,224,4Ø96,1Ø24,257
[R73]   117: LOOPZ,225,4Ø96,1Ø24,257
[VN2]   118: MOV,16Ø,3,128,16389
[XP2]   119: MOV,162,128,3,16389
[PM2]   12Ø: MOV,138,12,144,9221
[RN2]   121: MOV,136,144,12,1Ø29
[BG2]   122: MOV,184,8,768,4161
[FF2]   123: MOV,176,4,256,4161
[6N2]   124: MOV,142,64,144,9217
[EN2]   125: MOV,14Ø,144,96,1Ø25
[7V2]   126: MOVB,198,144,256,4225
[9U2]   127: MOVSB,164,1Ø24,1Ø24,1
[NV2]   128: MOVSW,165,1Ø24,1Ø24,1
[KW2]   129: MOVW,199,144,768,4225
[9F2]   13Ø: MUL,246,3,144,8357
[ZK2]   131: NEG,246,12,1Ø24,157
[MP2]   132: NEGB,246,144,1Ø24,153
[AR2]   133: NEGW,247,144,1Ø24,153
[AL2]   134: NOP,144,1Ø24,1Ø24,1
[PN2]   135: NOT,246,12,1Ø24,149
[9T2]   136: NOTB,246,144,1Ø24,145
[ZT2]   137: NOTW,247,144,1Ø24,145
[X22]   138: OR,12,3,768,41Ø1
[XA2]   139: OR,128,12,768,4237
[X12]   14Ø: OR,8,144,12,1Ø29
[X42]   141: OR,1Ø,12,144,9221
[AP2]   142: ORB,128,144,256,4233
[Y42]   143: ORG,2,768,1Ø24,Ø
[RT2]   144: ORW,129,144,768,4233
[8V1]   145: OUT,238,8,1,1
[GV1]   146: OUT,239,8,2,1
[EF2]   147: OUT,23Ø,256,3,2Ø53
[F62]   148: POP,88,8,1Ø24,65
[3P2]   149: POP,143,144,1Ø24,129
[D62]   15Ø: POP,7,64,1Ø24,65
[7N2]   151: POPF,157,1Ø24,1Ø24,1
[8M2]   152: PROC,5,12288,1Ø24,Ø
[RA2]   153: PUSH,8Ø,8,1Ø24,65
[BU2]   154: PUSH,255,144,1Ø24,177
[AA2]   155: PUSH,6,96,1Ø24,65
[NU2]   156: PUSHF,156,1Ø24,1Ø24,1
[553]   157: RCL,2Ø8,12,17408,&H8Ø95
[Ø93]   158: RCLB,2Ø8,144,17408,&H8Ø91
[P93]   159: RCLW,2Ø9,144,17408,&H8Ø91
[T73]   16Ø: RCR,2Ø8,12,17408,&H8Ø9D
[8A3]   161: RCRB,2Ø8,144,17408,&H8Ø99
[YA3]   162: RCRW,2Ø9,144,17408,&H8Ø99
[KK2]   163: REP,243,1Ø24,1Ø24,1
[7M2]   164: REPE,243,1Ø24,1Ø24,1
[WS2]   165: REPNE,242,1Ø24,1Ø24,1
[6U2]   166: REPNZ,242,1Ø24,1Ø24,1
[WM2]   167: REPZ,243,1Ø24,1Ø24,1
[KL2]   168: RET,2Ø3,1Ø24,8192,1
[BM2]   169: RET,195,1Ø24,4Ø96,1
[7V2]   17Ø: RET,2Ø2,768,8192,4Ø97
[1W2]   171: RET,194,768,4Ø96,4Ø97
[663]   172: ROL,2Ø8,12,17408,&H8Ø85
[3A3]   173: ROLB,2Ø8,144,17408,&H8Ø81
[TA3]   174: ROLW,2Ø9,144,17408,&H8Ø81
[U83]   175: ROR,2Ø8,12,17408,&H8Ø8D
```

```
[BB3]   176: RORB,2Ø8,144,17408,&H8Ø89
[1C3]   177: RORW,2Ø9,144,17408,&H8Ø89
[4L2]   178: SAHF,158,1Ø24,1Ø24,1
[T53]   179: SAL,2Ø8,12,17408,&H8ØA5
[493]   18Ø: SALB,2Ø8,144,17408,&H8ØA1
[U93]   181: SALW,2Ø9,144,17408,&H8ØA1
[G83]   182: SAR,2Ø8,12,17408,&H8ØBD
[DA3]   183: SARB,2Ø8,144,17408,&H8ØB9
[3B3]   184: SARW,2Ø9,144,17408,&H8ØB9
[672]   185: SBB,28,3,768,41Ø1
[AK2]   186: SBB,128,12,768,4253
[HA2]   187: SBB,24,144,12,1Ø29
[ØB2]   188: SBB,26,12,144,9221
[AR2]   189: SBBB,128,144,256,4249
[NS2]   19Ø: SBBW,129,144,768,4249
[YP2]   191: SCASB,174,1Ø24,1Ø24,1
[AS2]   192: SCASW,175,1Ø24,1Ø24,1
[872]   193: SEG,38,96,1Ø24,65
[AZ3]   194: SEGMENT,1Ø,&HFFFF,&HFFFF,Ø
[E63]   195: SHL,2Ø8,12,17408,&H8ØA5
[T93]   196: SHLB,2Ø8,144,17408,&H8ØA1
[FA3]   197: SHLW,2Ø9,144,17408,&H8ØA1
[293]   198: SHR,2Ø8,12,17408,&H8ØAD
[1B3]   199: SHRB,2Ø8,144,17408,&H8ØA9
[RB3]   2ØØ: SHRW,2Ø9,144,17408,&H8ØA9
[5L2]   2Ø1: STC,249,1Ø24,1Ø24,1
[MK2]   2Ø2: STD,253,1Ø24,1Ø24,1
[ØL2]   2Ø3: STI,251,1Ø24,1Ø24,1
[1U2]   2Ø4: STOSB,17Ø,1Ø24,1Ø24,1
[DV2]   2Ø5: STOSW,171,1Ø24,1Ø24,1
[R82]   2Ø6: SUB,44,3,768,41Ø1
[YM2]   2Ø7: SUB,128,12,768,4269
[2C2]   2Ø8: SUB,4Ø,144,12,1Ø29
[HC2]   2Ø9: SUB,42,12,144,9221
[XS2]   21Ø: SUBB,128,144,256,4265
[6S3]   211: SUBTTL,7,&HFFFF,&HFFFF,Ø
[8U2]   212: SUBW,129,144,768,4265
[SM2]   213: TEST,168,3,768,41Ø1
[5S2]   214: TEST,246,12,768,4229
[1P2]   215: TEST,132,144,12,1Ø29
[UP2]   216: TEST,132,12,144,9221
[1Z2]   217: TESTB,246,144,256,4225
[613]   218: TESTW,247,144,768,4225
[VL3]   219: TITLE,7,&HFFFF,&HFFFF,Ø
[VL2]   22Ø: WAIT,155,1Ø24,1Ø24,1
[DD2]   221: XCHG,144,8,1Ø24,65
[G82]   222: XCHG,144,2,8,8257
[CM2]   223: XCHG,134,144,12,1Ø29
[5N2]   224: XCHG,134,12,144,9221
[HL2]   225: XLAT,215,1Ø24,1Ø24,1
[BA2]   226: XOR,52,3,768,41Ø1
[KE2]   227: XOR,48,144,12,1Ø29
[5E2]   228: XOR,5Ø,12,144,9221
[HP2]   229: XOR,128,12,768,4277
[KU2]   23Ø: XORB,128,144,256,4273
[XV2]   231: XORW,129,144,768,4273
[TG1]   232: AX,192,Ø,26
[ØH1]   233: CX,193,8,24
[CK1]   234: DX,194,16,24
[8K1]   235: BX,195,24,24
[KK1]   236: SP,196,32,24
[GH1]   237: BP,197,4Ø,24
[GK1]   238: SI,198,48,24
[KH1]   239: DI,199,56,24
[1F1]   24Ø: AL,192,Ø,21
[HT1]   241: CL,193,8,164Ø4
[4H1]   242: DL,194,16,2Ø
[ØH1]   243: BL,195,24,2Ø
[KG1]   244: AH,196,32,2Ø
[PG1]   245: CH,197,4Ø,2Ø
[BH1]   246: DH,198,48,2Ø
[7H1]   247: BH,199,56,2Ø
[W72]   248: [BX+SI],Ø,Ø,16
[Ø72]   249: [BX+DI],1,Ø,16
[Ø72]   25Ø: [BP+SI],2,Ø,16
[462]   251: [BP+DI],3,Ø,16
[7P1]   252: [SI],4,Ø,16
[XM1]   253: [DI],5,Ø,16
[TP1]   254: [BX],7,Ø,16
[T41]   255: ES,Ø,Ø,64
[651]   256: CS,Ø,8,32
[PB1]   257: SS,Ø,16,64
[LA1]   258: DS,Ø,24,64
[2R1]   259: NEAR,Ø,Ø,4Ø96
[CN1]   26Ø: FAR,Ø,Ø,8192
```

Listing 4. Die Datei »sam3v11.dat« enthält die Daten, die SAM III für die Arbeit benötigt

Der imaginäre Drucker

Die Umleitung des Datenstroms der Standardein- und -ausgabe ist unter MS-DOS sehr leicht zu bewerkstelligen. Mit den Anweisungen »<« und »>« (Input/Output Redirection) lassen sich die meisten Aufgaben lösen. Eine sehr interessante Anwendung aber wird von MS-DOS nicht unterstützt: die Umleitung von Druckerdaten in eine Datei.

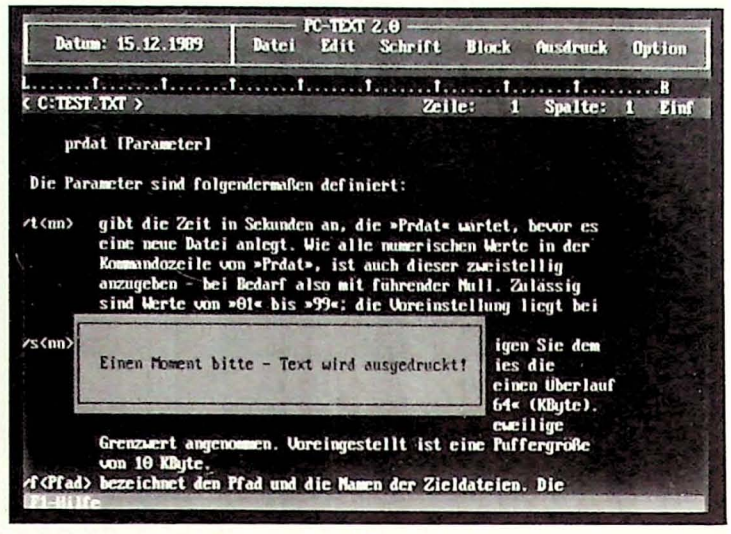
Zur Umleitung des Datenflusses stellt das Betriebssystem die sogenannten »Pipes« zur Verfügung. Dabei dient das Pipe-Zeichen »<« zur Umleitung des standardmäßigen Eingabekanals, während »>« den entsprechenden Ausgabekanal unter MS-DOS umlenkt. Wenn Sie aber Daten, anstatt diese an den Drucker zu schicken, in eine Datei umleiten wollen, dann läßt Sie das Betriebssystem im Stich. Schade, denn eine Druckdatenumleitung brächte viele Vorteile: Weil das Daten liefernde Programm nicht auf einen langsamen Drucker warten muß, läßt sich der Druckvorgang – insbesondere beim Ausdruck von Grafiken – erheblich beschleunigen. Die entstandene Datei drucken Sie bei einer späteren Gelegenheit (oder dann, wenn Sie ohnehin eine Kaffeepause einlegen) mit

```
copy print.dat /b prn
```

oder unter Zuhilfenahme eines Spoolers (zum Beispiel »Print«) aus. Zudem kommt es Nachbarn und Familienmitgliedern sehr entgegen, wenn ihre Nachtruhe nicht durch den kreischenden Nadeldrucker eines Computerbegeisterten gestört wird. In all diesen Fällen schafft das Programm »prdat.asm« (Listing 1 und 2) Abhilfe.
Nach dem Aufruf von Prdat installiert sich das Programm resident im Arbeitsspeicher Ihres Rechners. Es leitet dann alle Zeichen in Dateien um, die über den Interrupt 17hex an den Drucker gesandt werden. Bei einem erneuten Aufruf deinstalliert sich Prdat vollständig; das heißt, es gibt den vorher beanspruchten Speicherplatz wieder frei. Der Aufruf hat das Format

```
prdat [Parameter]
```

wobei die Parameter folgendermaßen definiert sind:
– **/t<nn>** gibt die Zeit in Sekunden an, die Prdat wartet, bevor es eine neue Datei anlegt. Wie alle numerischen Werte in der Kommandozeile von Prdat, ist auch dieser zweistellig anzugeben – bei Bedarf also mit führender Null. Zulässig sind Werte von »01« bis »99«; die Voreinstellung liegt bei fünf Sekunden.
– **/s<nn>** legt die Größe des Puffers in KByte fest. Billigen Sie dem Puffer möglichst viel Arbeitsspeicher zu, weil dies die Arbeitsgeschwindigkeit von Prdat erhöht und einen Überlauf verhindert. Zulässig sind Werte von »05« bis »64« (KByte). Bei kleineren oder größeren Angaben wird der jeweilige Grenzwert angenommen. Voreingestellt ist eine Puffergröße von 10 KByte.
– **/f<Pfad>** bezeichnet den Pfad und die Namen der Zieldateien. Die Angabe ist mit einem Punkt abzuschließen. Folgt daran anschließend eine Dateierweiterung, so wird diese ignoriert, weil Prdat selbst die Erweiterung »001« bis »999« hinzufügt. Einen Pfad sollten Sie immer dann angeben, wenn ein Programm in andere Unterverzeichnisse wechselt. Prdat legt neue Dateien stets im aktuellen Unterverzeichnis an und würde die Dateien daher im gesamten Verzeichnisbaum verteilen. Voreingestellt ist der Dateiname »prdat.(001)«.



Für den Ausdruck formatierte Daten werden in einer Datei gespeichert und erst bei Bedarf an den Drucker gesendet

– **/p** bewirkt, daß neben der Umleitung in Dateien, die Ausgabe der Daten zusätzlich auf dem Drucker erfolgt.
– **/?** hat ebenso wie jedes andere Zeichen hinter dem Schrägstrich die Ausgabe eines Hilfstextes zur Folge. Eine Installation von Prdat erfolgt in diesem Fall nicht.
Das folgende Beispiel zeigt die Verwendung und Bedeutung der Parameter:

```
prdat /t03 /s48 /fc:\editor\texte\brief. /p
```

Diese Kommandozeile veranlaßt Prdat, vor dem Anlegen einer neuen Datei drei Sekunden zu warten. Der Puffer hat eine Größe von 48 KByte, und die Daten werden in die Dateien »brief.001« bis »brief.999« in das Unterverzeichnis »editor\texte« auf dem Laufwerk C geschrieben. Zusätzlich erfolgt die Ausgabe auf dem Drucker.
Was aber ist nun der Grund dafür, daß es eines Programms wie Prdat bedarf, um die Umleitung von Druckerdaten zu realisieren, obwohl die Umleitung mit Hilfe des Kommando-Interpreters »command.com« in den anderen Fällen reibungslos funktioniert? Das Betriebssystem MS-DOS »versagt« hier aus mehreren Gründen: Zum einen liegt es daran, daß viele Anwenderprogramme den Drucker nicht über MS-DOS-Routinen ansprechen, zum anderen aber auch daran, daß eine Datenumleitung auf Betriebssystemebene nicht immer sinnvoll ist. »copy abc.txt prn« bei gleichzeitiger Umleitung in eine Datei ist ziemlicher Unsinn, weil »copy abc.txt print.dat« den ganzen Vorgang ohne »Klimmzüge« abwickelt. Eine Umleitung der Ausgabedaten ist also nur dort sinnvoll, wo ein Text oder eine Grafik nicht nur zum Drucker kopiert, sondern formatiert gedruckt werden soll. Dies aber ist Anwenderprogrammen vorbehalten und keine Aufgabe des Betriebssystems oder des Kommandointerpreters.
An dieser Stelle kommt der Interrupt 17hex ins Spiel. Dieser BIOS-Interrupt erlaubt mit drei Funktionen das komfortable Ansteuern des Druckers:
– Funktion 0 druckt ein Zeichen
– Funktion 1 initialisiert den Drucker
– Funktion 2 ermittelt den Druckerstatus
Dabei ist jeweils die Nummer des Druckers (das heißt die Nummer der parallelen Schnittstellen) im Register DX, die Funktionsnummer in AH und bei Funktion 0 das zu druckende Zeichen im Register AL zu übergeben.
An diesem Interrupt kommt kein Programm – auch MS-DOS nicht – vorbei; es sei denn, es übernimmt die Ansteuerung der parallelen Schnittstelle in Eigenregie. Letzteres ist jedoch recht selten, womit die Strategie von Prdat feststeht. Das Programm muß sich speicherresident installieren, den Interrupt 17hex abfangen und das zu druckende Zeichen in eine Datei schreiben.

Dies hört sich recht einfach an – ist es aber nicht! MS-DOS hat die unangenehme Eigenschaft, daß es nicht »reentrant« ist. Diese Unfähigkeit zum »Wiedereintritt«, zum rekursiven Aufruf von MS-DOS-Funktionen, entsteht aus folgender Eigenheit: Eine aufgerufene MS-DOS-Routine verwendet einen internen Speicherbereich (Stapelspeicher oder »Stack«) zur Ablage temporär benötigter Werte. Wird nun eine weitere MS-DOS-Funktion aufgerufen, während sich die erste noch in der Bearbeitung befindet, greift auch diese zweite auf den »privaten« Stack zu. Dabei werden die von der ersten Routine abgelegten Werte überschrieben. Ist die zweite Funktion beendet, wird die Bearbeitung der ersten fortgesetzt. Diese geht davon aus, daß sich die zuvor gesicherten Daten immer noch unverändert im Stapelspeicherbereich befinden und arbeitet fortan mit den falschen Werten weiter.
Sollte dies ausnahmsweise gutgehen, so folgt der »Crash« jedoch unweigerlich beim Beenden des Interrupts. Die für den Rücksprung benötigte Adresse befand sich nämlich ebenfalls auf dem privaten Stack von MS-DOS. Weil auch sie von der zweiten Routine zerstört wurde, wird die Interrupt-Routine mit einem Sprung ins Ungewisse beendet, und das geht immer schief.

Druckerdaten werden erst zwischengespeichert und dann in eine Datei geschrieben

Zur Lösung des Problems bieten sich zwei Wege an: Das Programm könnte die Dateiverwaltung selbst übernehmen und auf die Verwendung von Betriebssystemfunktionen verzichten oder sich die zu druckenden Zeichen so lange merken, bis es wieder möglich ist, MS-DOS aufzurufen. Die erste Lösung verbietet sich von selbst, weil Prdat ein Tool und kein Betriebssystem sein soll. Bleibt Lösung Nummer zwei, deren erster Teil sich leicht bewerkstelligen läßt.
Prdat legt dazu einen Puffer an, dessen Größe Sie in der Kommandozeile angeben. In ihm werden die ankommenden Zeichen zwischengespeichert. Durch die Nutzung dieses Puffers schlagen Sie zwei Fliegen mit einer Klappe: Zum einen hat Prdat nun mehr Zeit, um auf die Bereitschaft von MS-DOS zu warten, und zum anderen muß nicht für jedes Zeichen ein Dateizugriff erfolgen. Dies macht das Programm sehr viel schneller und komfortabler.
Doch wie soll Prdat feststellen, ob MS-DOS-Funktionen zur Zeit aufgerufen werden dürfen oder nicht? Die Lösung stellt MS-DOS selbst zur Verfügung. Den undokumentierten Interrupt 28hex, der als »Scheduler« (Steuerprogramm) bezeichnet wird, ruft MS-DOS immer dann auf, wenn es aktiv, aber dennoch bereit ist, einen Befehl zu bearbeiten. Dies ist beispielsweise dann der Fall, wenn der Kommando-Interpreter auf Eingaben wartet.
Prdat muß also auch diesen Interrupt auf eine eigene Routine legen, die bei Bedarf den vollen Puffer in die Datei schreibt. Es ist aber zu beachten, daß das Steuerprogramm nur diejenigen Funktionen des Interrupt 21hex aufrufen darf, die nicht ihrerseits den Scheduler aktivieren, weil es sonst zu einer Endlosschleife mit Stack-Überlauf käme. In diese Kategorie fallen die Funktionen bis zur Nummer 0Ehex. Für Prdat spielt das keine Rolle, weil die Funktionen für die Dateiverwaltung alle über 3Chex liegen.
Leider ist dies aber noch nicht der Weisheit letzter Schluß. Anwenderprogramme tun Ihnen nämlich nicht den Gefallen, dafür zu sorgen, daß MS-DOS den Scheduler aktiviert – jedenfalls nicht oft genug. Prdat benötigt also eine weitere Möglichkeit, den Betriebszustand von MS-DOS zu erfragen. Wieder hilft Ihnen das Betriebssystem selbst weiter – und zwar erneut mit einer undokumentierten Funktion. Die Rede ist vom »DOS Critical Interval Flag«, das den Wert Null hat, wenn sich MS-DOS in einem »unkritischen« Zustand befindet. Die Adresse dieses Statusbytes liefert die Funktion 34hex des Interrupt 21hex im Registerpaar ES:BX zurück.
Eine Routine, die dieses Flag möglichst oft kontrolliert und gegebenenfalls den Speichervorgang einleitet, würde die Sicherheit von Prdat erheblich erhöhen. Bevor nun aber der Inhalt des Puffers in eine Datei gespeichert wird, muß sich der Interrupt 1Chex noch beim Interrupt-Controller »abmelden«.
All diese Vorkehrungen gewährleisten jedoch noch immer nicht, daß der Puffer bei Bedarf sofort gespeichert wird. Er benötigt daher einen Toleranzbereich, der bei Prdat 3 KByte beträgt. Ist der Puffer also bis auf 3 KByte Restkapazität gefüllt, unternimmt Prdat den Versuch, den Pufferinhalt zu speichern. Das Programm hat dann noch weitere 3x1024 Zeichen lang Zeit, dies zu tun, bevor ein Überlauf erfolgt.
Ein Überlauf kann aber dennoch eintreten; und zwar dann, wenn MS-DOS selbst druckt – oder besser gesagt kopiert – und daher zu keinem Zeitpunkt angesprochen werden darf. Diesen Fall provozieren Sie beispielsweise mit

```
copy abc.txt prn
```

sofern die Datei »abc.txt« größer als die Kapazität des Puffers ist. Einfache Texteditoren (zum Beispiel die Editoren der Turbo-Programmiersprachenreihe) verwenden häufig die Dateifunktionen des Betriebssystems, um Texte aus dem Arbeitsspeicher an den Drucker zu senden. Wenn Sie daher die Größe des Puffers so wählen, daß die gesamte Datei hineinpaßt, gibt es keine Probleme. Die maximal zur Verfügung stehenden 64 KByte sollten dabei ausreichen, weil dies meist auch die maximale Textgröße der erwähnten Editoren ist. Andernfalls setzen Sie im Quelltext des Programms die Grenze (border) herunter, erweitern dadurch den Toleranzbereich auf mehr als 3 KByte, so daß Prdat mehr Zeit zum Speichern bleibt.
Sollte es trotz allem zu einem Überlauf kommen, besitzt Prdat eine diesbezügliche Sicherheitssperre. Über das programminterne Statusbyte »pr_flag«, das eine wichtige Rolle im Programm spielt, erfolgt die Anweisung, keine weiteren Zeichen mehr entgegenzunehmen. Die nachfolgenden Zeichen gehen daher ins Leere. Ist der Druckvorgang beendet, wird der Pufferinhalt gesichert, wobei die Datei dann allerdings unvollständig ist. Nur so läßt sich aber ein Absturz des Rechners und ein damit verbundener vollständiger Datenverlust vermeiden.
Nach den bisherigen Überlegungen wäre Prdat nur in der Lage, alle Druckerdaten in eine einzige Datei zu schreiben. Das ist noch sehr unbefriedigend, weil zum Beispiel fünf in einer »Sitzung« geschriebene Briefe in einer einzigen Datei recht unpraktisch sind. Es gilt also, eine Möglichkeit zu finden, mit der sich feststellen läßt, wann ein Druckvorgang beginnt und wann er endet. Leider gibt es weder von Seiten des Betriebssystems noch von Seiten der Software-Hersteller einen »Standarddruckvorgang«, der diese Informationen übermittelt.
Wie so oft bringt ein kleiner Trick den gewünschten Erfolg. Ihm liegt folgende Überlegung zugrunde: Während eines Druckvorgangs wird der Interrupt 17hex oft und vor allem schnell hintereinander aufgerufen. Zwischen zwei Druckvorgängen aber wird etwas mehr Zeit verstreichen. Sie benötigen daher nur eine Routine zur Zeitmessung, die diese Differenz ermittelt. Damit aber ist Prdat bereits ausgestattet.
Es ist der periodische Interrupt 1Chex, der auf eine Programmroutine von Prdat umgelenkt wurde. Diese Routine inkrementiert nun einen Zähler, der bei jedem Aufruf des Interrupt 17hex auf den Wert Null zurückgesetzt wird. Erreicht er dennoch einen bestimmten Grenzwert, dann zeigt dies eine Druckpause an. Prdat geht dann davon aus,

daß ein Druck beendet wurde und der nächste Aufruf des Interrupt 17hex einen neuen Druckvorgang einleitet. Es muß nur noch der Rest aus dem Puffer in die alte Datei gesichert und beim nächsten Druck eine neue Datei angelegt werden.
Die Zeitdauer, die Prdat warten soll, läßt sich über die Kommandozeile verändern. Damit ist eine Anpassung an Programme gewährleistet, die beim Drucken längere Pausen einlegen.

Die logische Verknüpfung aller Ideen und Informationen – das Programm Prdat

Ein kleines Problem stellt nur noch die logische Verkettung und Verarbeitung all dieser Informationen in einem Programm dar. Im folgenden wird daher der Aufbau von Prdat etwas näher erläutert.
Nach dem Start verzweigt Prdat in die Prozedur »installation«, an deren Anfang zunächst überprüft wird, ob sich das Programm bereits im Arbeitsspeicher befindet. Ist dies der Fall, so erfolgt die Deinstallation durch Restaurieren der alten Interrupt-Vektoren und das Freigeben des Speichers.
Soll Prdat dagegen installiert werden, so wird zuerst die Kommandozeile interpretiert, die im PSP (Program Segment Prefix) ab Offset 80hex steht. Findet das Programm dort Parameter, dann übernehmen die Prozeduren »set_time«, »set_path« und »set_size« die Umwandlung der Parameter und legen ihre Ergebnisse in den Variablen »maxti«, »filename« und »nameend« beziehungsweise »buf_size« ab.
Die alten Interrupt-Vektoren werden mit der Funktion 35hex des Interrupt 21hex ermittelt und in den Variablen »old_int??« abgelegt. Daraufhin erfolgt das Umlenken der Interrupts auf die neuen Routinen. Anschließend errechnet das Programm das Segment des Puffers (buffer), den Speicherplatzbedarf des gesamten Programms und den Wert für »border« (als Offset relativ zum Segment »buffer«).
Dieses Bereitstellen eines weiteren Segments bezeichnet man als »Relokation«. Dies ist eine Möglichkeit, die 64-KByte-Grenze von Com-Programmen zu überschreiten. Im Falle von »prdat.com« erlaubt diese Methode die Verwendung eines 64 KByte großen Puffers zusätzlich zum Speicherbedarf des Programms selbst. Der benötigte Speicher wird mit Funktion 31hex (Keep Process) reserviert und das Programm anschließend resident beendet.
Ab diesem Zeitpunkt sind die drei Interrupt-Prozeduren aktiv und bereit, die Umleitung durchzuführen. Erfolgt ein Aufruf des Interrupt 17hex mit Funktion 0 (Zeichen drucken), dann wird dieses im Puffer abgelegt. Ist die Schwelle »border« erreicht, setzt die Int-17hex-Routine das Bit »to_write« im Statusbyte »pr_flag«.
Dieses Bit beeinflußt die beiden Prozeduren »new_int1c« und »new_int28«. Stellen Sie bei ihrem Aufruf fest, daß es gesetzt ist, wird der Schreibvorgang mit Hilfe der Prozedur »write« durchgeführt. Die Routine des Interrupt 1Chex muß zuvor noch das »DOS Critical Interval Flag« abfragen, um sicher zu sein, daß ein MS-DOS-Zugriff erfolgen darf. Außerdem überprüft die Prozedur noch, ob »counter« den Wert »maxti« erreicht hat, und setzt in diesem Fall die Bits »to_write« und »last_time«.
Diese Bitkombination veranlaßt die Write-Prozedur dazu, bei ihrem nächsten Aufruf einen neuen Dateinamen zu erzeugen und die beiden Bit »new_dat« und »to_wait« zu setzen. Das letztgenannte Bit bewirkt, daß die Routinen des Interrupt 1C und 28hex erst nach dem nächsten Aufruf des Interrupt 17hex wieder aktiv werden. Dieses Vorgehen verhindert, daß Prdat fortwährend neue Dateien erzeugt, obwohl gar nicht gedruckt wird. Das Bit »new_dat« leitet nach dem nächsten Aufruf des Interrupt 17hex das Bereitstellen einer neuen Datei ein.
Prdat verhält sich in jeder Beziehung wie ein Drucker. Das bedeutet, daß es auf eine Statusabfrage oder Initialisierung (Funktion 1 und 2 des Interrupt 17hex) dem aufrufenden Programm die erwarteten Rückmeldungen gibt. Es bedeutet aber auch, daß Prdat im Druckerstatusbyte einen Fehler meldet, wenn beispielsweise beim Speichern des Pufferinhalts ein solcher aufgetreten ist.
Diese Lösung ist sicherer als das Verschweigen von Fehlern. Die meisten Anwendungsprogramme reagieren nämlich auf Druckerfehler in der Art, daß dem Anwender kein Schaden oder Datenverlust entsteht. Sie haben in diesem Fall die Möglichkeit, das Programm zu verlassen und Prdat erneut zu installieren. Dies wäre aber nicht gewährleistet, wenn Prdat fehlerhaft weiterarbeiten würde. Druckerfunktionen, die Prdat unbekannt sind, werden kommentarlos weitergeleitet. Weil das Programm den Drucker ersetzt, können Sie auch dann »drucken«, wenn keiner angeschlossen ist.
Geben Sie allerdings die Option »/p« beim Aufruf des Programms an, dann haben diese Aussagen nur begrenzte Gültigkeit. Prdat verhält sich nun nämlich nicht mehr wie ein Drucker, sondern leitet nach der Bearbeitung alle Signale an den Drucker weiter. Es erfolgen also keine Rückmeldungen durch das Programm. Tritt jedoch ein Fehler der oben beschriebenen Art auf, dann meldet ihn Prdat in jedem Fall und unterbricht zusätzlich die Verbindung zum Drucker.

(David Mehlhausen/zi)

Name	prdat.asm
Sprache	MASM ab Version 4.0, TASM
Grafikkarte	jede
Besonderheiten	keine

```
[T64]    1: ; Programmname: PRDAT.ASM
[4E3]    2: ; (C)  Everts & Hagedorn
[8G3]    3: ;      DOS International
[GX4]    4: ;.....................................
[0Y2]    5: ; Assemblieren mit
[3V3]    6: ;  Microsoft Assembler:
[PA2]    7: ;     MASM PRDAT;
[AB2]    8: ;     LINK PRDAT;
[1E4]    9: ;     EXE2BIN  PRDAT PRDAT.COM
[P30]   10: ;
[G33]   11: ;  Turbo Assembler:
[G92]   12: ;     TASM PRDAT
[XU2]   13: ;     TLINK PRDAT /T
[GX4]   14: ;.....................................
```

```
[TF4]   15: CODE         SEGMENT PARA 'CODE'
[A61]   16:              ORG 100h
[PM6]   17: ASSUME  CS:CODE, DS:CODE, ES:CODE, SS:CODE
[GX4]   18: ;.....................................
[9G4]   19: START:       jmp INSTALLATION
[NF6]   20: ;=========================================
[000]   21:
[GC5]   22: ;************ DEFINITIONEN ***************
[SU3]   23: ;Bitmasken für PR_FLAG
[8U3]   24: TO_WRITE        equ 00000001b
[306]   25: ;Puffer muß geschrieben werden
[CN3]   26: NEW_DAT         equ 00000010b
[NW5]   27: ;Neue Datei muß erzeugt werden
[UT3]   28: LAST_TIME       equ 00000100b
[A16]   29: ;Neue Datei, nachdem Rest geschrieben
[LW3]   30: PRINT_TOO       equ 00001000b
[W45]   31: ;Daten auch an den Drucker senden
[U93]   32: STOP            equ 00010000b
[X76]   33: ;Kurz stoppen, da Puffer restlos voll
[0S3]   34: TO_WAIT         equ 00100000b
[TA5]   35: ;Datei erst nach INT 17h erzeugen
[K93]   36: BUSY            equ 01000000b
[F16]   37: ;Es erfolgt gerade ein Dateizugriff
[4C3]   38: ERROR           equ 10000000b
```

```
[WG6]   39: ;Fehler, Programm endgültig stoppen
[E45]   40: ;----------------- MAKROS -----------------
[8X4]   41: PUSHM                   MACRO REGISTERS
[1H3]   42:         IRP REG,<REGISTERS>
[4E1]   43:             push REG
[HPØ]   44:         ENDM
[HPØ]   45: ENDM
[7E2]   46: ;---------------------
[NT4]   47: POPM                    MACRO REGISTERS
[1H3]   48:         IRP REG,<REGISTERS>
[WB1]   49:             pop REG
[HPØ]   50:         ENDM
[HPØ]   51: ENDM
[NF6]   52: ;=========================================
[ØØØ]   53:
[YT4]   54: ;*************** DATEN ********************
[HW2]   55: OLD_INT17       dd (?)
[9X2]   56: OLD_INT1C       dd (?)
[MW2]   57: OLD_INT28       dd (?)
[NK3]   58: ;Interrupt-Vektoren
[7E2]   59: ;---------------------
[FW2]   60: DOS_STAT        dd (?)
[ØN6]   61: ;Zeiger auf DOS-CRITICAL-INTERVAL-Flag
[7E2]   62: ;---------------------
[9T5]   63: FILENAME        db "PRDAT.ØØ1", 119 dup (Ø)
[B77]   64: ;128 Zeichen ist sehr großzügig bemessen
[CM5]   65: NAMEEND         dw (offset FILENAME + 8)
[M17]   66: ;Offset für das letzte Zeichen des Namens
[7E2]   67: ;---------------------
[EW2]   68: MAXTI           dw (5*18)
[FS6]   69: ;ca. 5 Sekunden als Standardeinstellung
[7E2]   70: ;---------------------
[ZH4]   71: BUF_ADR         equ this dword
[EV2]   72: POINTER         dw (Ø)
[FN2]   73: BUFFER          dw (?)
[5R5]   74: ;POINTER ist Offs., BUFFER ist Seg.
[7E2]   75: ;---------------------
[BP2]   76: BORDER          dw (?)
[1T3]   77: BUF_SIZE        dw (1Ø * 1Ø24)
[1R4]   78: ;1Ø KByte als Voreinstellung
[7E2]   79: ;---------------------
[S16]   80: PR_FLAG         db (NEW_DAT or TO_WAIT)
[WP6]   81: ;Datei erst nach INT-17h-Aufruf erzeugen
[7E2]   82: ;---------------------
[4W2]   83: COUNTER         dw (Ø)
[6E2]   84: ;Zeitzähler
[NF6]   85: ;=========================================
[ØØØ]   86:
[GA5]   87: ;************ PROZEDUREN ******************
[AL3]   88: NEWNAME             PROC NEAR
[TC1]   89:                 push bx
[LD2]   90:                 mov bx,NAMEEND
[RD3]   91:                 cmp byte ptr [bx],"9"
[WZ1]   92:                 je TO_ZERO3
[9Ø3]   93:                 inc byte ptr [bx]
[173]   94:                 jmp short LEAVE_NN
[ED5]   95: TO_ZERO3:       mov byte ptr [bx],"Ø"
[SV3]   96:                 cmp byte ptr [bx-1],"9"
[SZ1]   97:                 je TO_ZERO2
[GD3]   98:                 inc byte ptr [bx-1]
[173]   99:                 jmp short LEAVE_NN
[WU5]  100: TO_ZERO2:       mov byte ptr [bx-1],"Ø"
[VV3]  101:                 cmp byte ptr [bx-2],"9"
[ZF1]  102:                 jne INC_3
[XR2]  103:                 or PR_FLAG,ERROR
[2T5]  104: ;Wenn Erweiterung > 999, dann ERROR
[EW4]  105: INC_3:          inc byte ptr [bx-2]
[VY2]  106: LEAVE_NN:       pop bx
[3PØ]  107:                 ret
[6S2]  108: NEWNAME             ENDP
[SN5]  109: ;Dateierweiterung inkrementieren
[2S2]  110: ;-------------------------
[TN3]  111: CREATEDAT           PROC NEAR
[K32]  112:                 PUSHM <ax,cx>
[UM1]  113:                 mov ah,3Ch
[PH1]  114:                 xor cx,cx
[FT3]  115:                 mov dx,offset FILENAME
[381]  116:                 int 21h
[232]  117:                 jnc LEAVE_CD
[XR2]  118:                 or PR_FLAG,ERROR
[P26]  119: LEAVE_CD:       and PR_FLAG,(not NEW_DAT)
[B12]  120:                 POPM <cx,ax>
[3PØ]  121:                 ret
[NU2]  122: CREATEDAT           ENDP
[ML2]  123: ;Datei erzeugen
[2S2]  124: ;-------------------------
[YA3]  125: WRITE               PROC NEAR
[4M3]  126:                 PUSHM <ax,bx,cx,dx,es>
[TK4]  127:                 and PR_FLAG,(not TO_WRITE)
[6N3]  128: REP_WR:         mov ax,3D81h
[FT3]  129:                 mov dx,offset FILENAME
[381]  130:                 int 21h
[FB4]  131: ;Bestehende Datei öffnen
[M12]  132:                 jc WR_ERROR
[CH1]  133:                 mov bx,ax
[TK2]  134: ;Handle nach bx
[HV1]  135:                 mov ax,42Ø2h
[PH1]  136:                 xor cx,cx
[TH1]  137:                 xor dx,dx
[381]  138:                 int 21h
[LT5]  139: ;Dateizeiger auf das Ende der Datei
[M12]  140:                 jc WR_ERROR
[TH1]  141:                 xor dx,dx
[BC1]  142:                 push ds
[E64]  143:                 lds cx,dword ptr BUF_ADR
[NL1]  144:                 mov ah,4Øh
[381]  145:                 int 21h
[XY4]  146: ;Puffer in die Datei schreiben
[E31]  147:                 pop ds
[M12]  148:                 jc WR_ERROR
[9A2]  149:                 mov POINTER,Ø
[YM1]  150:                 mov ah,3Eh
[381]  151:                 int 21h
[564]  152: ;Datei wieder schließen
[M12]  153:                 jc WR_ERROR
[3Ø4]  154:                 test PR_FLAG,LAST_TIME
[SU1]  155:                 jz ONLY_WR
[K12]  156:                 call NEWNAME
[YØ8]  157:                 xor PR_FLAG,(TO_WAIT or NEW_DAT
                             or LAST_TIME)
[YH6]  158: ;Wenn LAST_TIME, dann neuen Dateinamen
[2Z6]  159: ;erzeugen und NEW_DAT und TO_WAIT setzen
[AR5]  160: ONLY_WR:        and PR_FLAG,(not STOP)
[V95]  161: LEAVE_WR:       POPM <es,dx,cx,bx,ax>
[3PØ]  162:                 ret
[U15]  163: WR_ERROR:       test PR_FLAG,STOP
[TE1]  164:                 jz ENDERR
[K12]  165:                 call NEWNAME
[FD2]  166:                 call CREATEDAT
[MU3]  167:                 and PR_FLAG,(not STOP)
[HW1]  168:                 jmp REP_WR
[ØY6]  169: ;Wenn ERROR und STOP, erst Datei erzeugen
[8D4]  170: ENDERR:         or PR_FLAG,ERROR
[Z73]  171:                 jmp short LEAVE_WR
[UF2]  172: WRITE               ENDP
[NF6]  173: ;=========================================
[5Z4]  174: ;*************** INT 17h ******************
[NK3]  175: NEW_INT17           PROC FAR
[813]  176:                 jmp short ROUTINE
[GX4]  177: ;.....................................
[791]  178:                 db "DMPD"
[3D6]  179: ;Die Kennung für die Deinstallation
[GX4]  180: ;.....................................
[2S2]  181: ROUTINE:       pushf
[XG3]  182:                 cmp cs:PR_FLAG,ERROR
[4P1]  183:                 jb FOR_ME?
[H82]  184:                 mov ah,1ØØ11ØØØb
[LSØ]  185:                 popf
[GRØ]  186:                 iret
[ØØ6]  187: ;Wenn ERROR, dann zurück mit ERROR
[MX2]  188: FOR_ME?:       or ah,ah
[T22]  189:                 jz NEW_CHAR
[BN4]  190:                 test cs:PR_FLAG,PRINT_TOO
[3A2]  191:                 jz NOT_PRINT1
[LSØ]  192:                 popf
[G83]  193:                 jmp cs:[OLD_INT17]
[U37]  194: ;Wenn ah <> Ø und PRINT_TOO, dann alten INT
[BC3]  195: NOT_PRINT1: cmp ah,1
[RV1]  196:                 jne NO_INI
[F82]  197:                 mov ah,ØØØ1ØØØØb
[PØ3]  198: NO_INI:        cmp ah,2
[ZV1]  199:                 jne NO_ASK
[G82]  200:                 mov ah,1ØØ1ØØØØb
[8H2]  201: NO_ASK:        popf
[GRØ]  202:                 iret
[PN6]  203: ;Sonst mit jeweiliger Nachricht zurück
[B15]  204: NEW_CHAR:       PUSHM <ds,es,di,cs>
[E31]  205:                 pop ds
[LB4]  206:                 and PR_FLAG,(not TO_WAIT)
[T13]  207:                 test PR_FLAG,STOP
[ØY1]  208:                 jnz LEAVE17
[654]  209:                 les di,dword ptr BUF_ADR
[ULØ]  210:                 cld
[NXØ]  211:                 stosb
[LG3]  212: ;Puffer beschreiben
[CK2]  213:                 mov POINTER,di
[1T2]  214:                 cmp di,BUF_SIZE
[MR1]  215:                 jb IN_SIZE
[SN2]  216:                 or PR_FLAG,STOP
[EP4]  217: ;Wenn Puffer voll, dann STOP
[814]  218: IN_SIZE:        cmp di,BORDER
[CV1]  219:                 jne LEAVE17
[FF3]  220:                 or PR_FLAG,TO_WRITE
[S46]  221: ;Wenn BORDER erreicht, dann TO_WRITE
[LW3]  222: LEAVE17:        mov COUNTER,Ø
[ZL2]  223:                 POPM <di,es,ds>
[BN4]  224:                 test cs:PR_FLAG,PRINT_TOO
[5A2]  225:                 jz NOT_PRINT2
[LSØ]  226:                 popf
[G83]  227:                 jmp cs:[OLD_INT17]
[ZØ6]  228: ;Wenn PRINT_TOO, dann alten INT 17h
[SC4]  229: NOT_PRINT2: mov ah,11Ø1ØØØØb
[LSØ]  230:                 popf
[GRØ]  231:                 iret
[DV4]  232: ;Sonst mit Nachricht zurück
[YW2]  233: NEW_INT17           ENDP
[NF6]  234: ;=========================================
[9Ø5]  235: ;*************** INT 1Ch ******************
[DL3]  236: NEW_INT1C           PROC FAR
[ZXØ]  237:                 pushf
```

Listing 1. Mit »prdat.asm« werden Sie von Ihrem Drucker unabhängig

```
[AZ3]  238:                cmp cs:PR_FLAG,TO_WAIT
[FF1]  239:                jae IMM1C
[TC6]  240: ;Wenn PR_FLAG >= TO_WAIT, dann zurück
[AK3]  241:                PUSHM <ax,ds,es,di,cs>
[E31]  242:                pop ds
[W32]  243:                inc COUNTER
[952]  244:                mov ax,MAXTI
[RK2]  245:                cmp COUNTER,ax
[B42]  246:                jne IN_TIME
[MZ5]  247:                or PR_FLAG,(TO_WRITE or LAST_TI
       ME)
[CC7]  248: ;Wenn Zeit überschritten, dann Flags setzen
[V34]  249: IN_TIME:       cmp PR_FLAG,0
[4K1]  250:                je LEAVE1C
[ED4]  251:                les di,dword ptr DOS_STAT
[3U3]  252:                cmp byte ptr es:[di],0
[WW1]  253:                jne LEAVE1C
[C35]  254: ;Wenn DOS-Flag > 0, dann zurück
[ZL1]  255:                mov al,20h
[3W1]  256:                out 0A0h,al
[PU1]  257:                out 020h,al
[6T3]  258: ;Sonst INT 1Ch abmelden
[TP2]  259:                or PR_FLAG,BUSY
[Y42]  260: ;BUSY setzen
[YT3]  261:                test PR_FLAG,TO_WRITE
[Z01]  262:                jnz C1
[SH3]  263:                test PR_FLAG,NEW_DAT
[3M1]  264:                jz LEAVE1C
[FD2]  265:                call CREATEDAT
[LX2]  266:                jmp short LEAVE1C
[LX2]  267: C1:           call WRITE
[835]  268: ;Schreiben oder Datei erzeugen
[ZC5]  269: LEAVE1C:      and PR_FLAG,(not BUSY)
[SR6]  270: ;BUSY löschen und INT weiter bearbeiten
[2Z2]  271:                POPM <di,es,ds,ax>
[582]  272: IMM1C:        popf
[893]  273:                jmp cs:[OLD_INT1C]
[PX2]  274: NEW_INT1C        ENDP
[NF6]  275: ;=========================================
[AZ4]  276: ;************* INT 28h ******************
[SK3]  277: NEW_INT28        PROC FAR
[ZX0]  278:                pushf
[AZ3]  279:                cmp cs:PR_FLAG,TO_WAIT
[UE1]  280:                jae IMM28
[632]  281:                PUSHM <ds,cs>
[E31]  282:                pop ds
[TP2]  283:                or PR_FLAG,BUSY
[YT3]  284:                test PR_FLAG,TO_WRITE
[211]  285:                jnz C2
[SH3]  286:                test PR_FLAG,NEW_DAT
[4L1]  287:                jz LEAVE28
[FD2]  288:                call CREATEDAT
[YW2]  289:                jmp short LEAVE28
[PX2]  290: C2:           call WRITE
[PB5]  291: LEAVE28:      and PR_FLAG,(not BUSY)
[037]  292: ;Schreiben oder Datei erzeugen und zurück
[E31]  293:                pop ds
[G72]  294: IMM28:        popf
[L83]  295:                jmp cs:[OLD_INT28]
[1X2]  296: NEW_INT28        ENDP
[NF6]  297: ;=========================================
[GB4]  298: INSTEND          equ this byte
[X04]  299: ;Ab hier steht der Puffer
[NF6]  300: ;=========================================
[000]  301:
[365]  302: ;*************** MELDUNGEN ****************
[MV5]  303: INSTALLM     db 13,10,"PRDAT V2.1  (C) DOS "
[L37]  304: db "International  Autor: David Mehlhausen"
[DV6]  305: db 13,10,"Erneuter Aufruf deinstalliert da"
[9B3]  306: db "s Programm",13,10,"$"
[WA6]  307: ENTFERNT     db 13,10,"PRDAT V2.1 wurde dei"
[ZF3]  308: db "nstalliert",13,10,"$"
[Y26]  309: INFO         db 13,10,"PRDAT [/tTIME] [/fFI"
[KB6]  310: db "LE] [/sSIZE] [/p]",13,10,10,"TIME (Sek"
[HX6]  311: db "unden) und SIZE (KByte) sind zweistell"
[FC6]  312: db "ig anzugeben;",13,10,"FILE enthält den"
[H07]  313: db " Dateinamen ohne Erweiterung, aber mit"
[HK3]  314: db " Punkt »."",13,10,10
[7Y6]  315: db "Bsp.: PRDAT /t03 /s32 /fc:\editor\text"
[FB6]  316: db "e. /p",13,10,10,"Voreingestellt sind 5"
[S06]  317: db " Sekunden, 10 KByte und »PRDAT."""
[CR1]  318: db 13,10,10,"$"
[NF6]  319: ;=========================================
[000]  320:
[UF5]  321: ;************** INSTALLATION **************
[S04]  322: SET_TIME             PROC NEAR
[FP6]  323: ;SET_TIME geht von zweistelliger Angabe,
[LZ6]  324: ;bei Bedarf auch mit führender Null aus.
[7W3]  325:                mov al,byte ptr [di+1]
[EK1]  326:                sub al,30h
[5N1]  327:                mov dl,0Ah
[N21]  328:                mul dl
[GF1]  329:                mov bl,al
[9W3]  330:                mov al,byte ptr [di+2]
[9T1]  331:                sub al,030h
[EB1]  332:                add al,bl
[8M1]  333:                mov dl,12h
[N21]  334:                mul dl
[571]  335:                or ax,ax
[WT1]  336:                jz ST_TIME
[K62]  337:                cmp ax,(99 * 18)
[HR1]  338:                ja ST_TIME
[E67]  339: ;Grenzwerte überschritten, dann Standard
[242]  340:                mov MAXTI,ax
[WH2]  341: ST_TIME:      ret
[M53]  342: SET_TIME             ENDP
[463]  343: ;-----------------------------
[R04]  344: SET_PATH             PROC NEAR
[B01]  345:                inc di
[6G1]  346:                mov si,di
[TF1]  347:                mov al,"."
[KF1]  348:                mov bl,cl
[XZ1]  349:                repne scasb
[7P5]  350: ;Der Punkt schließt den Pfad ab
[UB2]  351:                jcxz LEAVE_SP
[0E1]  352:                sub bl,cl
[LF1]  353:                mov cl,bl
[736]  354: ;In cl ist jetzt die Länge des Pfades
[2S3]  355:                mov di,offset FILENAME
[9M1]  356:                rep movsb
[AS5]  357: ;Dieser wird nach FILENAME kopiert
[XF1]  358:                mov al,"0"
[NX0]  359:                stosb
[NX0]  360:                stosb
[E01]  361:                inc al
[NX0]  362:                stosb
[MF1]  363:                xor al,al
[NX0]  364:                stosb
[193]  365: ;»"001"0« anhängen
[6Z0]  366:                dec di
[6Z0]  367:                dec di
[AF2]  368:                mov NAMEEND,di
[K55]  369: ;Offset der "1" in NAMEEND merken
[VH2]  370: LEAVE_SP:    ret
[L53]  371: SET_PATH             ENDP
[8N2]  372: ;-----------------------
[000]  373:
[A24]  374: SET_SIZE             PROC NEAR
[TP6]  375: ;SET_SIZE geht von zweistelliger Angabe,
[LZ6]  376: ;bei Bedarf auch mit führender Null aus.
[7W3]  377:                mov al,byte ptr [di+1]
[EK1]  378:                sub al,30h
[5N1]  379:                mov dl,0Ah
[N21]  380:                mul dl
[GF1]  381:                mov bl,al
[9W3]  382:                mov al,byte ptr [di+2]
[9T1]  383:                sub al,030h
[EB1]  384:                add al,bl
[R36]  385: ;in al ist jetzt die Größe in KByte
[1C1]  386:                cmp al,64
[N62]  387:                jae MAX_SIZE
[S81]  388:                cmp al,5
[VZ5]  389: ;5 KByte und 64 KByte sind Grenzwerte
[TR1]  390:                jb LO_SIZE
[KT1]  391:                mov bx,1024
[N31]  392:                mul bx
[2X2]  393:                mov BUF_SIZE,ax
[3P0]  394:                ret
[9D5]  395: MAX_SIZE:    mov BUF_SIZE,0FFFFh
[3P0]  396:                ret
[TD5]  397: LO_SIZE:     mov BUF_SIZE,(5 * 1024)
[3P0]  398:                ret
[673]  399: SET_SIZE             ENDP
[000]  400:
[KZ2]  401: ;****************************
[SD4]  402: INSTALLATION         PROC NEAR
[DW1]  403:                mov ax,3517h
[381]  404:                int 21h
[XM4]  405:                cmp word ptr es:[bx+2],"MD"
[432]  406:                jne INSTALL
[LN4]  407:                cmp word ptr es:[bx+4],"DP"
[432]  408:                jne INSTALL
[VE6]  409: ;Kennung gefunden, dann deinstallieren
[Y15]  410:                lds dx,dword ptr es:OLD_INT17
[CW1]  411:                mov ax,2517h
[381]  412:                int 21h
[P25]  413:                lds dx,dword ptr es:OLD_INT1C
[WX1]  414:                mov ax,251Ch
[381]  415:                int 21h
[125]  416:                lds dx,dword ptr es:OLD_INT28
[LW1]  417:                mov ax,2528h
[381]  418:                int 21h
[U06]  419: ;Interrupt-Vektoren zurücksetzen
[6H1]  420:                mov bx,es
[VT2]  421:                mov es,es:[2Ch]
[7M1]  422:                mov ah,49h
[381]  423:                int 21h
[4H1]  424:                mov es,bx
[7M1]  425:                mov ah,49h
[381]  426:                int 21h
[V86]  427: ;Environment und Speicher freimachen
[8C1]  428:                push cs
[E31]  429:                pop ds
[ZU3]  430:                mov dx,offset ENTFERNT
[3M1]  431:                mov ah,09h
[381]  432:                int 21h
[BW1]  433:                mov ax,4C00h
[381]  434:                int 21h
[346]  435: ;Deinstall-Meldung ausgeben und Ende
[M53]  436: INSTALL:     push cs
[G31]  437:                pop es
[0W1]  438:                mov si,080h
[7D3]  439:                cmp byte ptr [si],0
```

```
[YL1]  44Ø:                je NOPARAM
[TR6]  441: ;Keine Kommandozeile, dann Standardwerte
[932]  442:                mov cl,[si]
[6F1]  443:                xor ch,ch
[5V1]  444:                mov di,Ø81h
[2F5]  445: ;Die Kommandozeile beginnt bei 81h
[C33]  446: SEARCH:        mov al,"/"
[E66]  447: ;Damit muß jeder Parameter beginnen
[ULØ]  448:                cld
[XZ1]  449:                repne scasb
[P42]  45Ø:                jcxz NOPARAM
[FU5]  451: ;Kein Parameter oder kein weiterer
[932]  452:                PUSHM <cx,di>
[8C7]  453: ;Pos. und Anzahl für nächste Suche merken
[ØK3]  454:                mov al,byte ptr [di]
[LG1]  455:                cmp al,"T"
[YE1]  456:                je F_TIME
[LG1]  457:                cmp al,"t"
[YE1]  458:                je F_TIME
[RF1]  459:                cmp al,"F"
[ZD1]  46Ø:                je F_PATH
[RF1]  461:                cmp al,"f"
[ZD1]  462:                je F_PATH
[HG1]  463:                cmp al,"S"
[8F1]  464:                je F_SIZE
[HG1]  465:                cmp al,"s"
[8F1]  466:                je F_SIZE
[BG1]  467:                cmp al,"P"
[ØS1]  468:                je F_PRINT
[BG1]  469:                cmp al,"p"
[ØS1]  47Ø:                je F_PRINT
[UD1]  471:                jmp HELP
[KW6]  472: ;Wenn hinter "/" keine Anweisung, dann INFO
[EØ4]  473: F_TIME:        call SET_TIME
[ET2]  474:                jmp short CONTS
[EZ3]  475: F_PATH:        call SET_PATH
[ET2]  476:                jmp short CONTS
[E14]  477: F_SIZE:        call SET_SIZE
[ET2]  478:                jmp short CONTS
[HD5]  479: F_PRINT:       or PR_FLAG,PRINT_TOO
[AN4]  48Ø: ;Nur PRINT_TOO Flag setzen
[TK3]  481: CONTS:         POPM <di,cx>
[4S1]  482:                jmp SEARCH
[CN3]  483: NOPARAM:       mov ax,3517h
[381]  484:                int 21h
[PS4]  485:                mov word ptr OLD_INT17+2,es
[7B4]  486:                mov word ptr OLD_INT17,bx
[XX1]  487:                mov ax,351Ch
[381]  488:                int 21h
[UT4]  489:                mov word ptr OLD_INT1C+2,es
[BC4]  49Ø:                mov word ptr OLD_INT1C,bx
[MW1]  491:                mov ax,3528h
[381]  492:                int 21h
[VS4]  493:                mov word ptr OLD_INT28+2,es
[CB4]  494:                mov word ptr OLD_INT28,bx
[WK5]  495: ;Alte Interrupt-Vektoren merken
[WL1]  496:                mov ah,34h
[381]  497:                int 21h
[VL4]  498:                mov word ptr DOS_STAT+2,es
[7B4]  499:                mov word ptr DOS_STAT,bx
[7Y5]  5ØØ: ;Adresse des DOS-CRITICAL-Flag holen
[P24]  5Ø1:                mov dx,offset NEW_INT17
[CW1]  5Ø2:                mov ax,2517h
[381]  5Ø3:                int 21h
[644]  5Ø4:                mov dx,offset NEW_INT1C
[WX1]  5Ø5:                mov ax,251Ch
[381]  5Ø6:                int 21h
[X24]  5Ø7:                mov dx,offset NEW_INT28
[LW1]  5Ø8:                mov ax,2528h
[381]  5Ø9:                int 21h
[S65]  51Ø: ;Interrupt-Vektoren verbiegen
[8V3]  511:                mov dx,offset INSTALLM
[3M1]  512:                mov ah,Ø9h
[381]  513:                int 21h
[AV2]  514: ;Install-Meldung
[MF4]  515:                mov ax,offset CODE:INSTEND
[DA1]  516:                mov cl,4
[VF1]  517:                shr ax,cl
[4H1]  518:                mov bx,cs
[AD1]  519:                add ax,bx
[K11]  52Ø:                inc ax
[K11]  521:                inc ax
[182]  522:                mov BUFFER,ax
[G65]  523: ;Segment des Puffers errechnen
[UV2]  524:                mov ax,BUF_SIZE
[X72]  525:                sub ax,(3 * 1Ø24)
[X82]  526:                mov BORDER,ax
[PF4]  527: ;BORDER = BUF_SIZE - 3 KByte
[WV2]  528:                mov bx,BUF_SIZE
[9M1]  529:                mov cl,Ø4h
[XF1]  53Ø:                shr bx,cl
[AA3]  531: ;Größe des Puffers
[3M3]  532:                mov dx,offset INSTEND
[1G1]  533:                shr dx,cl
[GD1]  534:                add dx,bx
[S11]  535:                inc dx
[S11]  536:                inc dx
[GD7]  537: ;Puffer+residenter Teil= nötiger Speicher
[6V1]  538:                mov ax,31ØØh
[381]  539:                int 21h
[485]  54Ø: ;Restlichen Speicher freigeben
```

```
[3G4]  541: HELP:          mov dx,offset INFO
[3M1]  542:                mov ah,Ø9h
[381]  543:                int 21h
[RX1]  544:                mov ax,4C48h
[381]  545:                int 21h
[MK3]  546: INSTALLATION          ENDP
[NF6]  547: ;=========================================
[ØØØ]  548:
[ZØ2]  549: CODE           ENDS
[4K1]  55Ø:                END START
```

Listing 1. Mit »prdat.asm« werden Sie von Ihrem Drucker unabhängig

```
[125]    1: E9 88 Ø4 ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[U45]    2: ØØ ØØ ØØ 5Ø 52 44 41 54 2E 3Ø 3Ø 31 ØØ ØØ ØØ ØØ
[ØZ4]    3: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]    4: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]    5: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]    6: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]    7: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]    8: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]    9: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[L35]   1Ø: ØØ ØØ ØØ 1B Ø1 5A ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 28 22
[KH5]   11: ØØ ØØ 53 8B 1E 93 Ø1 8Ø 3F 39 74 Ø4 FE Ø7 EB 2Ø
[NR5]   12: C6 Ø7 3Ø 8Ø 7F FF 39 74 Ø5 FE 47 FF EB 12 C6 47
[8S5]   13: FF 3Ø 8Ø 7F FE 39 75 Ø5 8Ø ØE 9F Ø1 8Ø FE 47 FE
[YH5]   14: 5B C3 5Ø 51 B4 3C 33 C9 BA 13 Ø1 CD 21 73 Ø5 8Ø
[AF5]   15: ØE 9F Ø1 8Ø 8Ø 26 9F Ø1 FD 59 58 C3 5Ø 53 51 52
[ZH5]   16: Ø6 8Ø 26 9F Ø1 FE B8 81 3D BA 13 Ø1 CD 21 72 41
[PL5]   17: 8B D8 B8 Ø2 42 33 C9 33 D2 CD 21 72 34 33 D2 1E
[8F5]   18: C5 ØE 97 Ø1 B4 4Ø CD 21 1F 72 26 C7 Ø6 97 Ø1 ØØ
[WG5]   19: ØØ B4 3E CD 21 72 1A F6 Ø6 9F Ø1 Ø4 74 Ø8 E8 71
[FN5]   2Ø: FF 8Ø 36 9F Ø1 26 8Ø 26 9F Ø1 EF Ø7 5A 59 5B 58
[WP5]   21: C3 F6 Ø6 9F Ø1 1Ø 74 ØD E8 57 FF E8 84 FF 8Ø 26
[TX5]   22: 9F Ø1 EF EB A1 8Ø ØE 9F Ø1 8Ø EB DF EB Ø4 44 4D
[8L5]   23: 5Ø 44 9C 2E 8Ø 3E 9F Ø1 8Ø 72 Ø4 B4 98 9D CF ØA
[6L5]   24: E4 74 1E 2E F6 Ø6 9F Ø1 Ø8 74 Ø6 9D 2E FF 2E Ø3
[GF5]   25: Ø1 8Ø FC Ø1 75 Ø2 B4 1Ø 8Ø FC Ø2 75 Ø2 B4 9Ø 9D
[FN5]   26: CF 1E Ø6 57 ØE 1F 8Ø 26 9F Ø1 DF F6 Ø6 9F Ø1 1Ø
[3R5]   27: 75 2Ø C4 3E 97 Ø1 FC AA 89 3E 97 Ø1 3B 3E 9D Ø1
[SK5]   28: 72 Ø5 8Ø ØE 9F Ø1 1Ø 3B 3E 9B Ø1 75 Ø5 8Ø ØE 9F
[AE5]   29: Ø1 Ø1 C7 Ø6 AØ Ø1 ØØ ØØ 5F Ø7 1F 2E F6 Ø6 9F Ø1
[AP5]   3Ø: Ø8 74 Ø6 9D 2E FF 2E Ø3 Ø1 B4 DØ 9D CF 9C 2E 8Ø
[5K5]   31: 3E 9F Ø1 2Ø 73 53 5Ø 1E Ø6 57 ØE 1F FF Ø6 AØ Ø1
[WD5]   32: A1 95 Ø1 39 Ø6 AØ Ø1 75 Ø5 8Ø ØE 9F Ø1 Ø5 8Ø 3E
[3G5]   33: 9F Ø1 ØØ 74 2B C4 3E ØF Ø1 26 8Ø 3D ØØ 75 21 BØ
[HH5]   34: 2Ø E6 AØ E6 2Ø 8Ø ØE 9F Ø1 4Ø F6 Ø6 9F Ø1 Ø1 75
[FV5]   35: ØC F6 Ø6 9F Ø1 Ø2 74 Ø8 E8 A7 FE EB Ø3 E8 BC FE
[SR5]   36: 8Ø 26 9F Ø1 BF 5F Ø7 1F 58 9D 2E FF 2E Ø7 Ø1 9C
[RG5]   37: 2E 8Ø 3E 9F Ø1 2Ø 73 24 1E ØE 1F 8Ø ØE 9F Ø1 4Ø
[SE5]   38: F6 Ø6 9F Ø1 Ø1 75 ØC F6 Ø6 9F Ø1 Ø2 74 Ø8 E8 71
[RY5]   39: FE EB Ø3 E8 86 FE 8Ø 26 9F Ø1 BF 1F 9D 2E FF 2E
[ØA5]   4Ø: ØB Ø1 ØD ØA 5Ø 52 44 41 54 2Ø 56 32 2E 31 2Ø 2Ø
[TC5]   41: 28 43 29 2Ø 44 4F 53 2Ø 49 6E 74 65 72 6E 61 74
[RD5]   42: 69 6F 6E 61 6C 2Ø 2Ø 41 75 74 6F 72 3A 2Ø 44 61
[XG5]   43: 76 69 64 2Ø 4D 65 68 6C 68 61 75 73 65 6E ØD ØA
[TA5]   44: 45 72 6E 65 75 74 65 72 2Ø 41 75 66 72 75 66 2Ø
[HG5]   45: 64 65 69 6E 73 74 61 6C 6C 69 65 72 74 2Ø 64 61
[6F5]   46: 73 2Ø 5Ø 72 6F 67 72 61 6D 6D ØD ØA 24 ØD ØA 5Ø
[5A5]   47: 52 44 41 54 2Ø 56 32 2E 31 2Ø 77 75 72 64 65 2Ø
[CH5]   48: 64 65 69 6E 73 74 61 6C 6C 69 65 72 74 ØD ØA 24
[HG5]   49: ØD ØA 5Ø 52 44 41 54 2Ø 5B 2F 74 54 49 4D 45 5D
[2H5]   5Ø: 2Ø 5B 2F 66 46 49 4C 45 5D 2Ø 5B 2F 73 53 49 5A
[5L5]   51: 45 5D 2Ø 5B 2F 7Ø 5D ØD ØA ØA 54 49 4D 45 2Ø 28
[VH5]   52: 53 65 6B 75 6E 64 65 6E 29 2Ø 75 6E 64 2Ø 53 49
[UD5]   53: 5A 45 2Ø 28 4B 42 79 74 65 29 2Ø 73 69 6E 64 2Ø
[1K5]   54: 7A 77 65 69 73 74 65 6C 6C 69 67 2Ø 61 6E 7A 75
[MF5]   55: 67 65 62 65 6E 3B ØD ØA 46 49 4C 45 2Ø 65 6E 74
[XD5]   56: 68 84 6C 74 2Ø 64 65 6E 2Ø 44 61 74 65 69 6E 61
[WF5]   57: 6D 65 6E 2Ø 6F 68 6E 65 2Ø 45 72 77 65 69 74 65
[FD5]   58: 72 75 6E 67 2C 2Ø 61 62 65 72 2Ø 6D 69 74 2Ø 5Ø
[CL5]   59: 75 6E 6B 74 2Ø AF 2E AE ØD ØA ØA 42 73 7Ø 2E 3A
[Y95]   6Ø: 2Ø 5Ø 52 44 41 54 2Ø 2F 74 3Ø 33 2Ø 2F 73 33 32
[PG5]   61: 2Ø 2F 66 63 3A 5C 65 64 69 74 6F 72 5C 74 65 78
[FF5]   62: 74 65 2E 2Ø 2F 7Ø ØD ØA ØA 56 6F 72 65 69 6E 67
[RC5]   63: 65 73 74 65 6C 6C 74 2Ø 73 69 6E 64 2Ø 35 2Ø 53
[YF5]   64: 65 6B 75 6E 64 65 6E 2C 2Ø 31 3Ø 2Ø 4B 42 79 74
[9H5]   65: 65 2Ø 75 6E 64 2Ø AF 5Ø 52 44 41 54 2E AE ØD ØA
[HL5]   66: ØA 24 8A 45 Ø1 2C 3Ø B2 ØA F6 E2 8A D8 8A 45 Ø2
[ZK5]   67: 2C 3Ø Ø2 C3 B2 12 F6 E2 ØB C9 74 Ø8 3D F6 Ø6 77
[AP5]   68: Ø3 A3 95 Ø1 C3 47 8B F7 BØ 2E 8A D9 F2 AE E3 19
[6Y5]   69: 2A D9 8A CB BF 13 Ø1 F3 A4 BØ 3Ø AA AA FE CØ AA
[6K5]   7Ø: 32 CØ AA 4F 4F 89 3E 93 Ø1 C3 8A 45 Ø1 2C 3Ø B2
[7N5]   71: ØA F6 E2 8A D8 8A 45 Ø2 2C 3Ø Ø2 C3 3C 4Ø 73 ØD
[DK5]   72: 3C Ø5 72 1Ø BB ØØ Ø4 F7 E3 A3 9D Ø1 C3 C7 Ø6 9D
[RM5]   73: Ø1 FF FF C3 C7 Ø6 9D Ø1 ØØ 14 C3 B8 17 35 CD 21
[DD5]   74: 26 81 7F Ø2 44 4D 75 45 26 81 7F Ø4 5Ø 44 75 3D
[DD5]   75: 26 C5 16 Ø3 Ø1 B8 17 25 CD 21 26 C5 16 Ø7 Ø1 B8
[3K5]   76: 1C 25 CD 21 26 C5 16 ØB Ø1 B8 28 25 CD 21 8C C3
[5P5]   77: 26 8E Ø6 2C ØØ B4 49 CD 21 8E C3 B4 49 CD 21 ØE
[YS5]   78: 1F BA DD Ø3 B4 Ø9 CD 21 B8 ØØ 4C CD 21 ØE Ø7 BE
[3R5]   79: 8Ø ØØ 8Ø 3C ØØ 74 4D 8A ØC 32 ED BF 81 ØØ B0 2F
[GR5]   8Ø: FC F2 AE E3 3F 51 57 8A Ø5 3C 54 74 1F 3C 74 74
[CH5]   81: 1B 3C 46 74 1C 3C 66 74 18 3C 53 74 19 3C 73 74
[ZL5]   82: 15 3C 5Ø 74 16 3C 7Ø 74 12 E9 99 ØØ E8 F3 FE EB
[AT5]   83: ØF E8 11 FF EB ØA E8 31 FF EB Ø5 8Ø ØE 9F Ø1 Ø8
[TM5]   84: 5F 59 EB BA B8 17 35 CD 21 8C Ø6 Ø5 Ø1 89 1E Ø3
[MG5]   85: Ø1 B8 1C 35 CD 21 8C Ø6 Ø9 Ø1 89 1E Ø7 Ø1 B8 28
[UL5]   86: 35 CD 21 8C Ø6 ØD Ø1 89 1E ØB Ø1 B4 34 CD 21 8C
[1N5]   87: Ø6 11 Ø1 89 1E ØF Ø1 BA 5C Ø2 B8 17 25 CD 21 BA
[DU5]   88: DD Ø2 B8 1C 25 CD 21 BA 3F Ø3 B8 28 25 CD 21 BA
[ØM5]   89: 72 Ø3 B4 Ø9 CD 21 B8 72 Ø3 B1 Ø4 D3 E8 8C CB Ø2
[LL5]   9Ø: C3 4Ø 4Ø A3 99 Ø1 A1 9D Ø1 2D ØØ ØC A3 9B Ø1 8B
[CP5]   91: 1E 9D Ø1 B1 Ø4 D3 EB BA 72 Ø3 D3 EA Ø3 D3 42 42
[5N5]   92: B8 ØØ 31 CD 21 BA ØØ Ø4 B4 Ø9 CD 21 B8 48 4C CD
[R7Ø]   93: 21
```

Listing 2. Das Hexdump-Listing zum Programm »prdat.com«

Impressum

Herausgeber
Christian Widuch

Leitung des Redaktionsbüros
Volker Everts (ev), Andreas Hagedorn (hg)

Chefredakteur
Martin Aschoff (ma)

Stellvertretende Chefredakteurin
Eva-Maria Hierlmeier (hi)

Redaktion
Alois Kneisle (al), Benjamin Kleinke (bk),
Werner Niedermaier (wn), Thomas Röder (tr),
Andrea Scheding (as), Johann Zieger (zi)

Schlußredaktion
Leitung: Andrea Voß
Ulrich Mayer, Karl-Heinz Peter, Markus Stein

Redaktionsassistenz
Claudia Beer, Susanne Bienert, Dorothea Lehner

Koordination
Stefan Ritter

Produktionsleitung
Gerd Köberich

Bereichsleitung
Claudia Ebbrecht (Fotosatz/Lektorat)
Margarete Schenk, Helmut Skoupy (Montage/Reprografie)

Layout
Yvonne Hendricks

Fotografie
Klaus Jatho

Fotosatz
Silvia Führer

Lektorat
Susanne Lessinger

Montage/Reprografie
Manuela Eska, Andrea Gundlach

Werbegestaltung
Mohamed Hawa, Petra Küch

Anzeigenverkaufsleitung
Wolfgang Schnell

Anzeigenverkauf für PLZ 1,4,5
Gerlinde Rachow (0 56 51) 8 09 - 3 90
Wolfgang Brill (0 56 51) 8 09 - 3 80

Anzeigenverkauf für PLZ 2 + 3
DMV-Verlagsbüro Hamburg
Ohlsdorfer Straße 34, 2000 Hamburg 60
Sylvia Ehrenpfordt, Tel. (0 40) 46 12 33, Telefax: (0 40) 47 43 10

Anzeigenverkauf für PLZ 6 – 8
DMV-Verlagsbüro München
Zaunkönigweg 2c, 8000 München 82,
Telefon: (0 89) 4 39 10 87, Telefax: (0 89) 4 39 10 80
Leitung: Britta Fiebig
Anzeigenverkauf: Monika Schöbel, Jens Dhein, Peter Schätzle

Anzeigenverwaltung und Disposition
Andrea Giese, Karina Ehrlich, Beate Kranz

Anschrift Redaktion:
Redaktionsbüro Everts & Hagedorn
Redaktion DOS International,
Hans-Stießberger-Str. 3, 8013 Haar/München
Telefon: (0 89) 46 10 01 -0
Telefax: (0 89) 46 10 01-90

Anschrift Verlag:
DMV Daten und Medien-Verlag
Widuch GmbH & Co. KG
Fuldaer Straße 6, 3440 Eschwege
Telefon: (0 56 51) 8 09-0
Telefax: (0 56 51) 80 9-3 33

Vertrieb
Verlagsunion Erich Pabel-Arthur Moewig KG (VPM)
Friedrich-Bergius-Straße 20
6200 Wiesbaden

Druck
Druckerei Jungfer, 3420 Herzberg

Bezugspreise
DOS Extra erscheint viermal jährlich.
Einzelpreis DM 18, –, 145 öS, 18, – sfr.

Abonnementpreise
Die Preise verstehen sich grundsätzlich einschließlich Porto und Verpackung.
Inland: Bezugspreis 4 Ausgaben: DM 60, –
Europäisches Ausland: Bezugspreis 4 Ausgaben: DM 70, –
Außereuropäisches Ausland: Bezugspreis 4 Ausgaben: DM 80, –

Bankverbindungen:
Postscheck Frankfurt/M: Kto.-Nr.: 23043-608
Raiffeisenbank Eschwege:
BLZ: 522 603 85, Kto. Nr.: 245 7008

Die Abonnementbestellung kann innerhalb einer Woche nach Auftrag beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich widerrufen werden. Zur Wahrung der Frist reicht der Poststempel. Das Abonnement verlängert sich automatisch um 4 Ausgaben, wenn es nicht mindestens 6 Wochen vor Ablauf beim Verlag schriftlich gekündigt wird.

Manuskripte, Programme:
bitte an das Redaktionsbüro Everts & Hagedorn, Redaktion DOS International, Hans-Stießberger-Straße 3, 8013 Haar/München, senden.

Für unverlangt eingesandte Manuskripte und Datenträger sowie Fotos übernehmen der Verlag und die Redaktion keine Haftung. Die Zustimmung zum Abdruck wird vorausgesetzt.
Das Urheberrecht für veröffentlichte Manuskripte liegt ausschließlich beim Verlag. Nachdruck sowie Vervielfältigung oder sonstige Verwertung von Texten nur mit schriftlicher Genehmigung des Verlages. Für veröffentlichte Programme und Basteleien übernehmen wir weder Gewähr noch Haftung. Schaltungen und verwendete Bezeichnungen müssen nicht frei von gewerblichen Schutzrechten sein.
Namentlich gekennzeichnete Fremdbeiträge geben nicht in jedem Fall die Meinung der Redaktion wieder.

Die Inserenten

Bath Beach Software.....59

CPS Computertechnik.....2

DMV.....13, 25, 47, 57, 73, 79, 97, 105, 163

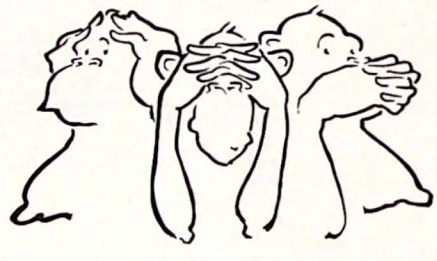
Interest Verlag.....34

Nürnberger PD-Studio.....101

Rubel.....101

Scholz.....101

Soft Maker.....164



TROPENWALD-VERNICHTUNG, WIESO? IST DOCH SO WEIT WEG.

Irrtum. Der tropische Regenwald reguliert auch unser Klima.

Der tropische Regenwald wird gnadenlos vernichtet. Jedes Jahr eine Fläche, die fast so groß wie die Bundesrepublik Deutschland ist. Mit katastrophalen Folgen für das Weltklima. Aber auch für die im Tropenwald lebenden Ureinwohner und Tier- und Pflanzenarten.

Wenn Sie wissen wollen, wie der tropische Regenwald gerettet werden kann, dann füllen Sie den Coupon aus. Wir beantworten Ihre Frage, sagen Ihnen, wo wir uns noch engagieren, warum Sie uns dabei helfen können.

Ja, ich will von ROBIN WOOD wissen, wie der tropische Regenwald gerettet werden kann.

Name, Vorname

Straße

Wohnort

Weil ich helfen will, lege ich 3 Mark in Briefmarken bei. Coupon einsenden an ROBIN WOOD, Postfach 10 21 22, 2800 Bremen 1

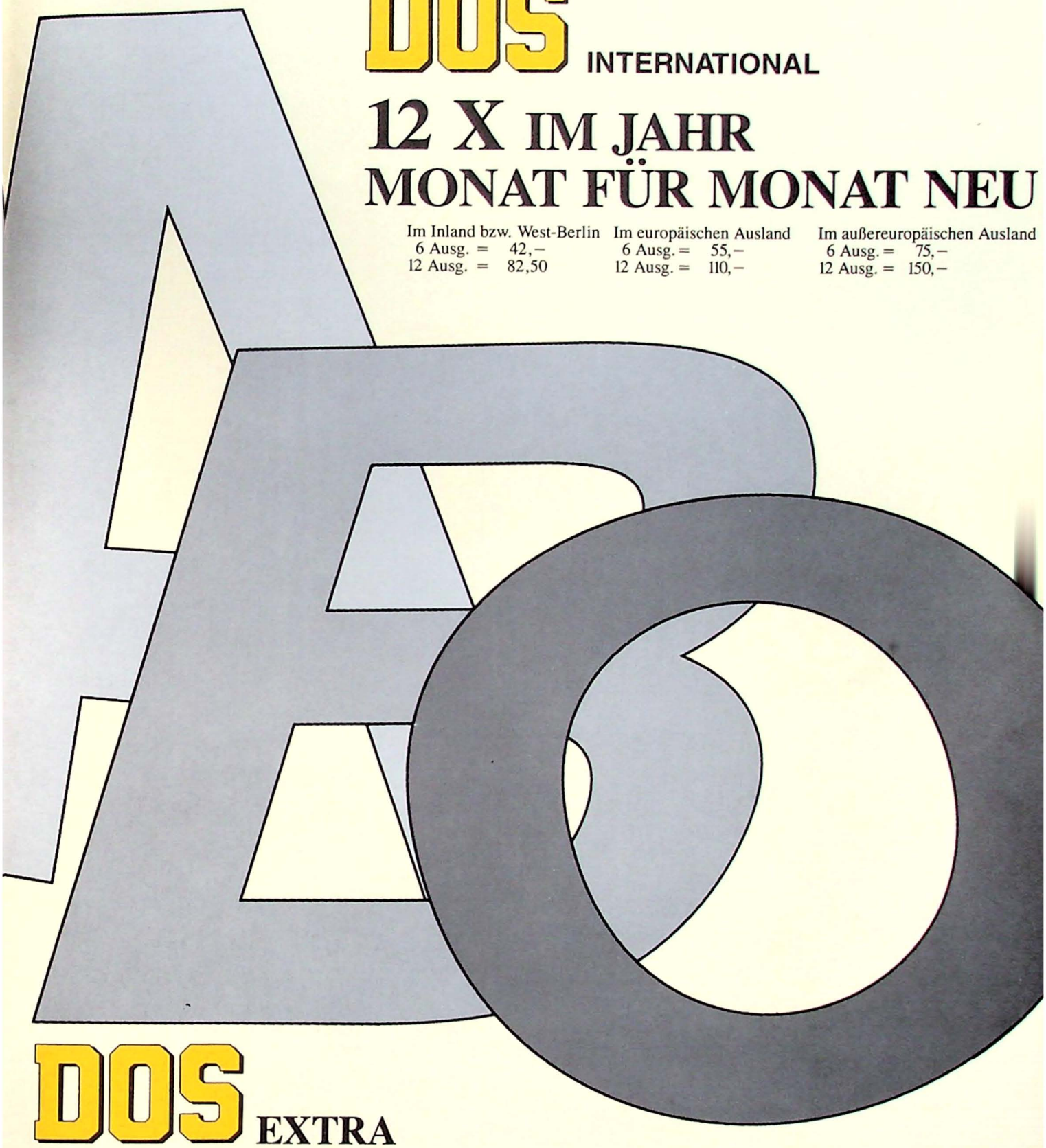
dmv ROBIN WOOD

Gewaltfreie Aktionsgemeinschaft für Natur und Umwelt e.V.

DOS INTERNATIONAL

12 X IM JAHR
MONAT FÜR MONAT NEU

Im Inland bzw. West-Berlin	Im europäischen Ausland	Im außereuropäischen Ausland
6 Ausg. = 42,–	6 Ausg. = 55,–	6 Ausg. = 75,–
12 Ausg. = 82,50	12 Ausg. = 110,–	12 Ausg. = 150,–



DOS EXTRA

4 X IM JAHR 4 X POWERPROGRAMME FÜR IHREN PC

Im Inland bzw. West-Berlin	Im europäischen Ausland	Im außereuropäischen Ausland
4 Ausgaben 60,– DM	4 Ausgaben 70,– DM	4 Ausgaben 80,– DM

Widerrufsrecht: Jeder Abonnent hat das Recht, seine Bestellung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich zu widerrufen. Die rechtzeitige Absendung des Widerrufsschreibens genügt zur Fristwahrung.

Bitte benutzen Sie die Bestellkarte.

DMV-VERLAG · POSTFACH 250 · 3440 ESCHWEGE

DMV Daten- und Medienverlag

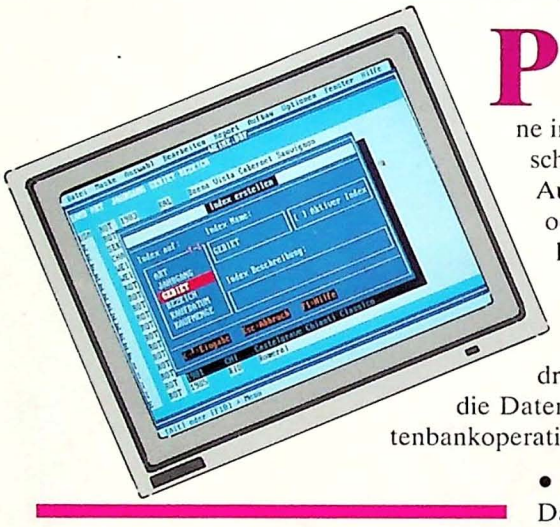
Neues zum Thema Datenverwaltung

Das organisierte Chaos. Zwanzig Ringordner, und keiner findet die Daten, die er braucht. Oder noch schlimmer: Zettelwirtschaft. Und das im EDV-Zeitalter...

Woran liegt's? Vielleicht an den hohen Preisen, die man bisher für so manche PC-Datenbank bezahlen mußte? Oder daran, daß man bei einigen Datenbanken erst einen Programmierkurs belegen muß, bevor man mit ihnen arbeiten kann?

DataMaker 1.0

räumt damit auf! Endlich gibt es ein erschwingliches Datenbankprogramm, das einen Leistungsvergleich mit vielen teuren Programmen nicht zu scheuen braucht und dazu noch extrem bedienerfreundlich ist.



Pulldown-Menüs nach dem **SAA**-Standard, perfekte Integration der Maus und übersichtliche Bildschirmfenster sorgen für eine intuitive Arbeitsumgebung, in der man sich schnell zurechtfindet.

Auf Knopfdruck öffnen Sie eine Datenbank oder erzeugen eine neue nach Ihren Vorstellungen. Schnell noch eine attraktive Bildschirmmaske kreiert, und die Dateneingabe kann beginnen.

Ebenso schnell erstellt man Berichte, druckt Adreßaufkleber, sortiert oder indiziert die Datenbestände und führt alle erdenklichen Datenbankoperationen mit Leichtigkeit durch.

• dBASE III Plus/dBASE IV-kompatible Datendateien • Bis zu 16 Indexdateien pro Datenbank • Direkter Datenaustausch mit TextMaker 2.2 • Import und Export von WKS-, WK1-, WordStar- und WORD-Datendateien • Memofelder • Integrierter Memofeld-Editor • Frei gestaltbare Bildschirmmaske • Tabellen- und Seitendarstellung • Sortieren nach beliebigen Feldern • Benutzerdefinierte Berichte für ansprechende Ausdrucke • Tastaturmakros • Integrierter Makrorecorder • Unterstützung für mehr als 600 Drucker.

DataMaker 1.0 erhalten Sie im Fachhandel für nur

DM 248,–*

* Unverbindliche Preisempfehlung

SoftMaker CeBIT '90 Halle 3, B 52

SoftMaker

SoftMaker Software GmbH · Kronacher Straße 7 · 8500 Nürnberg 90 · Telefon 09 11/30 10 49 · Fax 09 11/30 37 96

Coupon

☐ Schicken Sie mir bitte eine Demo-Diskette. 10-Mark-Schein liegt bei.

☐ 5.25" -Format

☐ 3.5" -Format

☐ DataMaker 1.0 ist genau, was ich suche! Schicken Sie mir ein Exemplar.

☐ Verrechnungsscheck anbei

☐ Nachnahme

Name _____

Straße _____

Ort _____

Datum _____

Unterschrift _____